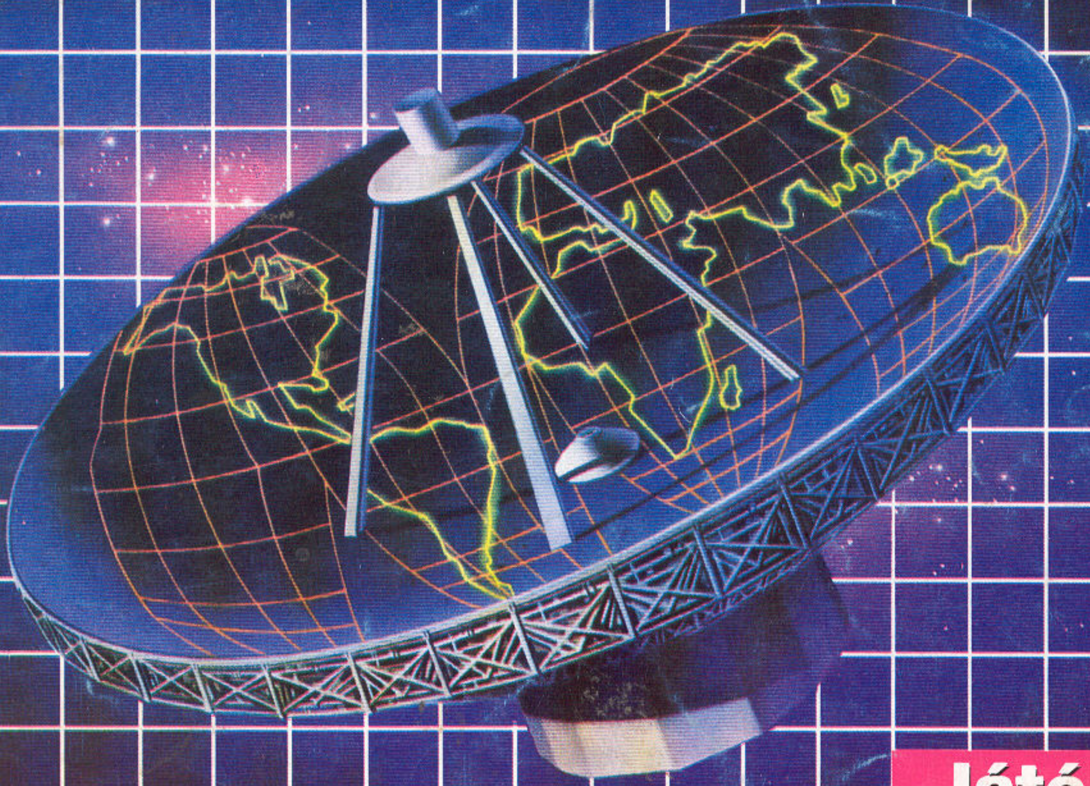
commodore újság

IX.évfolyam
1994/4
Az Országos Commodore Egyesület lapja

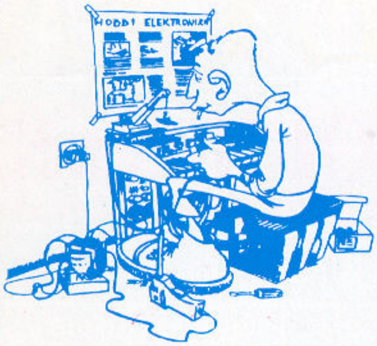
Szoftver-óra Commodore 64-re

Forrásprogramkészítő



Játékok

Grafikus iratkészítő



HOBBI ELEKTRONIKA

Szuper ajánlat!

Urbán István mérnök áramköreinek szaküzlete

Budapest VII., Dózsa György út 16. (Dózsa–Jobbágy sarok)
Nyitva: H–P 10–17-ig Tel./fax: 122-8892
(Zárás után üzenetrögzítő)

KÍNÁLATUNKBÓL

SZÁMÍTÁSTECHNIKA:

C64 BŐVÍTŐK	egys.	élesztve
PLOFI Datassette cartridge	1.200 Ft	1.500 Ft
PLOFI Help + cartridge	1.200 Ft	1.500 Ft
PLOFI FASTLOAD cartridge	1.200 Ft	1.500 Ft
PLOFI FASTLOAD + SPEEDTAPE	1.500 Ft	1.700 Ft
PLOFI SIMON'S cartridge	1.500 Ft	1.700 Ft
PLOFI JÁTÉK cartridge	1.500 Ft	1.700 Ft
ACTION REPLAY MK7 cartridge	—	3.450 Ft
ATOMIC POWER cartridge	—	3.450 Ft
FINAL III. cartridge	—	3.450 Ft
PAGEFOX DTP cartridge	3.200 Ft	3.900 Ft
MINI EPROMBANK cartridge	2.300 Ft	2.900 Ft
EPROMBANK 256 Kbájt	2.900 Ft	4.000 Ft
C64 DOKTOR teszt cartridge	2.300 Ft	3.000 Ft
EPROMÉGETŐ (2716-27512-ig)	3.400 Ft	4.500 Ft
IC TESZTER (kb. 180 TTL IC)	2.700 Ft	3.900 Ft
256K RAM BŐVÍTŐ RAM floppi	—	11.800 Ft
FÉNYCERUZA + szoftver	950 Ft	1.450 Ft
HANGDIGITALIZÁLÓ + szoftver	1.350 Ft	1.700 Ft
HANGKAPCSOLÓ + szoftver	700 Ft	990 Ft
DATASSETTE fejbeállító	390 Ft	—
CPU STOP + RESET	400 Ft	—
FREKVENCIAMÉRŐ 30 MHz	1.300 Ft	1.800 Ft
I/O BŐVÍTŐ 8255-tel	1.400 Ft	1.800 Ft
JOYSTICK automata	1.500 Ft	1.900 Ft
USER CENTRONICS kábel	750 Ft	—

IBM BŐVÍTŐK	egys.	élesztve
IBM IC TESZTER + szoftver	5.900 Ft	9.800 Ft
IBM EPROMÉGETŐ + szoftver	4.800 Ft	9.900 Ft
IBM EPROMÉGETŐ tápegység	1.200 Ft	↑ benne
IBM CODEC HANGKÁRTYA	3.800 Ft	4.800 Ft
IBM KOVOX	—	700 Ft
IBM AKTÍV KOVOX	1.600 Ft	—
IBM I/O KÁRTYA 48 vonal	3.500 Ft	—
IBM GAL PROGRAMOZÓ	6.500 Ft	8.500 Ft

TANGÓ PLD jogtiszta szoftver. *Kérjen árajánlatot!*
TANGÓ NYÁKTEVEZŐ jogtiszta szoftver
TTL, CMOS, TRANZISZTOR, DIÓDA, MEMÓRIA katalógus floppylemezen, egységesen: 600 Ft/db

C-64 BÖRZE

Használt C-64-et, flopit adok-veszek.
Hibás gépet beszámítok, megvásárolok.
Megegyezés szerint
új IBM PC részegységre cserélek.

Áraink a 25% ÁFA-t tartalmazzák!

Szaküzletemben vásárolhat a Commodore Újság HOBBI ELEKTRONIKA rovatában megjelent hardverleírásokhoz (működő minta alpján) panelt, egységcsomagot, vagy készre szerelt áramkört.

MŰSZERVÁSÁR
Bontott anyagok, használt műszerek böngészdéje



Állandóan változó készlet!

Vidéki olvasóknak segít a szerző levelező, egységcsomagküldő szolgáltatása: a megrendelt csomagot postán utánvéttel elküldöm. Telefonon/faxon és levélben is rendelhet.
A HOBBI ELEKTRONIKÁ-hoz nem kell hosszú levél. Rendelését néhány sorban, egyértelműen közölje.

Levélcím: 1656 Budapest Pf. 50.

Az üzletben megvásárolhatók az RT évkönyvei, a Rádiótechnika és a Hobby Elektronika korábbi számai.

SZERENCSEJÁTÉK RT

MIT, HOGYAN, HOL, MIKOR?

EGYESÜLETI ÜGYEK: Egyesületünknek tagja lehet mindenki, aki a tagsági díjat befizeti. A tagdíjat személyesen az egyesület irodjában (1025 Budapest, Vöröstorony utca 29. Telefon: 1-76-22-57), vagy átutalással az MNB 217-98 292, OTP 565-3610-8 számlára lehet befizetni. Megrendelés esetén számlát küldünk.

Pötyögőszolgálatunk valamint a szervizkedvezmény és az apróhirdetés lehetősége tagjaink rendelkezésére áll.

A DEÁKPÁHOLY tagjai minden hónapban megkapják a C-újságot, a tagsági díj egy évre 1020 Ft.

A PLUSZPÁHOLY tagjai minden hónapban megkapják a C-újságot, és kapnak havonta 3 db vásárlási utalványt. A tagsági díj egy évre 2350 Ft.

A SZUPERPÁHOLY tagjai havonta 15 példányt kapnak a C-újságból, és ezzel havonta 15×3 db vásárlási utalványt is. Az éves tagsági díj 24 000 Ft.

ÜGYFÉLFOGADÁS: Minden kedden és csütörtökön 12—16 óra között várjuk tagjainkat és az érdeklődőket.

PÖTYÖGŐSZOLGÁLAT: Az újságban megjelenő programokat másolja a megrendelők részére. Megrendelhető személyesen az egyesület irodájában vagy postai utánvéttel. Postacím: 1388 Budapest 62., Postafiók: 86.

APRÓHIRDETÉS: Az egyesületi tagoknak ingyen áll rendelkezésére. Nem tagoknak a hirdetés ára 300 Ft. A hirdetés módja: az újságban megjelenő nyomtatvány kitöltésével.

A C-ÚJSÁG RÉGEBBI SZÁMAI megvásárolhatók az egyesület irodájában, vagy megrendelhetők utánvéttel.

Kedvezményes ár! Tagoknak olcsóbb!

Az újságban eddig megjelent programok gépenként összegyűjtve megrendelhetők. VC 20, C16, PLUS/4, C128, C64. További felvilágosítást is adunk az 1-76-22-57-es telefonszámon vagy levélben!

Vidéki pluszpáholy-tagjaink háromhavi tikett összegyűjtésekor igénybe vehetik a NOVOTRADE 2C Áruház csomagküldő szolgálatát.

VIDÉKEN TOVÁBBI INFORMÁCIÓK KAPHATÓK:

Baja, AXIS Kft.
Győri Bartók Béla Művelődési Ház,
Jászberényi Városi Könyvtár,
Kecskemét, SZIGMA—BIT,
Pécsi Apáczai Csere János Gimnázium,
Zalaegerszegi Ságvári Endre Gimnázium.

Az Országos Commodore Egyesület módszertani kiadványa

Egyesületi iroda és szerkesztőség:
1025 Budapest, Vöröstorony utca 29. Telefon: 1-76-22-57
Felelős kiadó: Horváth Judit, az egyesület elnöke
Főszerkesztő: Rados Péter, az OCE főtitkára
Felelős szerkesztő: dr. Horváth András
Művészeti szerkesztő: Bausz Sándor
Levélcím: Commodore Újság, 1388 Budapest, 62. Pf.: 86.
Index: ISSN 0237-756 X
Terjeszti a Nemzeti Hírlapkereskedelmi Rt. és
a regionális részvénytársaságok
Megvásárolható a hírlapárusoknál
94.0071 MSZH Nyomda és Kiadó Kft., Budapest
Felelős vezető: Nagy László

FELHÍVÁS

Tisztelt Tagtársunk, Olvasóink!
Kérjük, hogy ha rendelkeznek olyan működő számítógéppel, vagy jó állapotban lévő bármilyen számítógép-tartozékkal, amire nincs szükségük, akkor hozzák el egyesületi irodánkba (minden héten kedden és csütörtökön 11—15 óra között).
Mi összegyűjtjük és eljuttatjuk azokat határainkon túli (erdélyi, kárpátaljai és szlovákiai) magyar iskolákhoz.
Köszönettel:

Országos Commodore Egyesület

C-64 C+4 AMIGA PC tulajdonosok!

Várunk benneteket az Országos Commodore Egyesület klubdélelőttjén a Havanna Közösségi Házban minden hónap 3. vasárnapján, délelőtt 9 órától. Gépet és hosszabítót hozzatok!
Cím: 1181 Budapest, Kondor Béla sétány 8.
Megközelíthető: a metró kőbánya–kispesti végállomásától a piros 136-os busszal.
Legközelebb:

április 17-én

Tisztelt Szerkesztőség!

Már régóta várom nagy érdeklődéssel lapjuk megjelenését az újságárusnál, mert mindig találni benne jó, érdekes programot.

Az 1993/7—8. számban leltem rá POZSGAI FERENC „TILI-TOLI"-jára, ami nagyon megtetszett kialakítása, működtetése miatt. Alkotója dicséretet érdemel, szép munkát végzett.

Azóta elég sokat „tilitoliztam" a programmal és felmerült bennem, hogy érdemes lenne egy csekély bővítést végrehajtani rajta.

Volt esetem, amikor hosszú időt próbálkoztam, nem sikerült az összerakás és feladással akartam befejezni, de megtudni az „elfecsérelt" időt is. Erre hoztam létre az „F" billentyűvel történő feladás lehetőségét.

Amikor pedig, gyakorlatot szerezve, sok sikeres próbálkozásom volt, szerettem volna tudni az összerakások idejének átlagát. Erre az esetre is elvégeztem a bővítést. Ezeket szeretném rendelkezésükre bocsájtani azzal a reménnyel, hogy ez másokat is érdekel és az eredeti szerzőt ezzel nem sértem meg.

Miután az átlagidő méréséhez szubrutint használtam, mellékelem az önállóan is működő időösszegző-átlagszámoló programot is. Hátha szüksége van valakinek erre is.

Cakó János

```
167 ifc$="f"thenprint"(clr)(down)(rght)feladtam...!!!":goto990

ready.
```

```
480 print"(down)(down)(wht)"tab(14)"time "left$(ti$,2)":"mid$(ti$,3,2)" "right$(
ti$,2):gosub1000

ready.
```

```
570 :
575 print"    A gep kirajzol egy 4x4-es mezot":print
580 print"    es elhelyez benne 15 betut,majd":print
585 print"    megkeveri.":print
590 print"    Feladat:":print
595 print"    A betuket ABC-sorrendbe kell rakni":print
600 print"    az ures hely mozgatasaval.":print:print
605 print"'F'-betuvel feladhato a jatek.":print
610 print"(lblu)(rvs)"tab(6)"Nyomjon le egy billentyut!"

ready.
```

```
745 gosub2000:end

ready.
```

```
990 print"(down)(down)(lblu)"tab(14)"time "left$(ti$,2)":"mid$(ti$,3,2)" "right$
(ti$,2)
992 print"(down)(rght)(rght)(rght)(rght)(rvs) a program 'run'-al ujra indithato
(off)"
995 end
997 :
998 rem ** az idok osszesitese **
999 :
1000 io$=left$(ti$,2):ip$=mid$(ti$,3,2):im$=right$(ti$,2)
1010 io=val(io$):ip=val(ip$):im=val(im$)
1020 oo=oo+io:op=op+ip:om=om+im
1030 s=s+1
1040 dec=oo+op/60+om/3600
1050 return
1997 :
1998 rem ** az atlag ido szamitasa **
1999 :
2000 ad=dec/(s-1):ao=int(ad)
2010 ap=int((ad-ao)*60)
2020 am=(((ad-ao)*60)-ap)*60
2030 print"(down)(down)(down)(rght)(rght)(rght)(rvs) jatekatlag (off) "ao":"ap":
"am
2040 print"(rght)(rght)(rght)(rght)(rvs) osszesen (off)(rght)(rght)"s-1"(rght)(r
ght)(rght)(rght)(rght)(rght)(rght)(rght)(rvs) jatekbol (off)"
2050 return

ready.
```

```
10 REM ******************************
20 REM *    IDOERTEKEK  ATLAGA      *
30 REM * ORA,PERC,MASODPERC ALAKBAN *
40 REM ******************************
50 :
80 CLR
90 S=S+1
100 INPUT"(DOWN) BEVITEL :ORA,PERC,MPERC ";OO,PP,MM
110 INPUT"(DOWN)(RGHT)(RVS) HELYES A BEVITEL  (RGHT)(RGHT)(RGHT)I(LEFT)(LEFT)(LEF
T)(OFF)";A$
120 IF A$="I" THEN 140
130 IF A$="N" THEN 100
140 O=O+OO
150 P=P+PP
160 M=M+MM
170 PRINT"(DOWN)"O"ORA",P"'",M CHR$(34)
180 PRINT"(UP)(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)
(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)(R
GHT)(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)ADAT:";S
190 INPUT"(RGHT)(RVS) TOVABB SZAMOLUNK  (RGHT)(RGHT)(RGHT)I(LEFT)(LEFT)(LEFT)(OFF
)";A$
200 IF A$="I" THEN 90
210 IF A$="N" THEN GOSUB 500:GOSUB 600:GOSUB700:END
497 :
498 REM * EREDMENYT SZAMITO SZUBRUTIN *
499 :
500 OM=INT((M/60-INT(M/60))*60+.5)
510 P=P+INT(M/60)
520 OP=INT((P/60-INT(P/60))*60+.5)
530 OO=O+INT(P/60)
540 PRINT"(DOWN)  (RVS)  AZ IDO OSSZESEN :(OFF)"
550 PRINT"(DOWN)"OO"ORA",OP"'",OM CHR$(34)
560 RETURN
597 :
598 REM * ATLAGOT SZAMITO SZUBRUTIN *
599 :
600 DEC=OO+OP/60+OM/3600
610 AD=DEC/S:AO=INT(AD):AP=INT((AD-AO)*60):AM=(((AD-AO)*60)-AP)*60
620 PRINT"(DOWN)  (RVS)  AZ ATLAG IDO    :(OFF)"
630 PRINT"(DOWN)"AO"ORA",AP"'",AM CHR$(34):PRINT
640 RETURN
697 :
698 REM * ELLENORZO SZUBRUTIN *
699 :
700 PRINTAD
710 PRINTAO+AP/60+AM/3600"(RGHT)(RGHT)(RVS) HA AZONOSAK : OK !!! (OFF)"
720 RETURN

READY.
```

Szoftver-óra a C-64-en

Mindenkinek szüksége lehet erre a kis segédprogramra, aki hajlamos arra, hogy programírás közben megfeledkezzen az időről. Ez ugyanis fenn a képernyő tetején állandóan mutatja a dátumot, a hét napját és a pontos időt mégpedig másodperc pontossággal. A betöltés és RUN-nal indítás után a program megkérdezi az aktuális adatokat. A hét napjának a nevét is beadva automatikusan indul, és a képernyő első sorában állandóan látszik. A program nem hajlítja el a normál megszakítási (interrupt) rutin mutatóit. Azokkal a programokkal nem viselik el egymást (nem kompatibilisak), amelyek használják a C000—C800 című memóriaterületet. A RESTORE billentyűvel a program az indítómenübe hozható. Itt újra beadhatók az indítási időadatok.

Hogyan működik?

— A program a C000-tól kezdődő memóriaterületre másolódik.

— Kérdezi a szükséges adatokat, mint pl. az időt stb.

— A megszakító rutin mutatóit beállítja.

— A CIA-n kívül a rendszermegszakítást is felhasználja.

— Azért, hogy a Basic-programokat se nagyon lassítsa le, a pontos időt csak a másodperc-adat megváltozásakor újítja fel.

— A feliratot közvetlenül a képernyő-memóriaterületre írja a 0400 címtől kezdődően.

SET-tel ki lehet szállni belőle, de mindaddig, amíg a memóriában van,

```
1 REM ********************************
2 REM *                              *
3 REM *         DATUM ES ORA         *
4 REM *                              *
5 REM ********************************
6 :
7 :
8 PRINT CHR$(147)"DATA BEOLVASAS ES ELLENORZES ...";:J=02049:VE=03522:P=J
9 FOR B=0 TO 7:READ A$
10 L=ASC(MID$(A$,2,1))
11 H=ASC(MID$(A$,1,1))
12 L=L-48:IF L>9 THEN L=L-7
13 H=H-48:IF H>9 THEN H=H-7
14 PRINT"■■■■■■■■■■■■"P;:P=P+1
15 IF H>15 OR L>15 THEN 17
16 A=H*16+L:POKE J+B,A:T=T+A:NEXT B:READ A:IF A=T THEN 18
17 PRINT:PRINT"DATA HIBA ...  SOR:"PEEK(64)*256+PEEK(63):END
18 T=0:J=J+8:IF J<VE THEN 9
19 PRINT"■■KESZ":END
20 DATA 45,08,C7,07,9E,32,31,31,0589
21 DATA 39,20,20,20,20,43,49,41,0390
22 DATA 32,2D,49,4E,54,45,52,52,0563
23 DATA 55,50,54,55,48,52,20,20,0552
24 DATA 20,20,20,20,20,20,56,4F,0357
25 DATA 4E,20,41,4E,44,52,45,41,0537
26 DATA 53,20,48,4F,45,53,53,4C,0577
27 DATA 45,52,20,43,48,45,4D,4E,0546
28 DATA 49,54,5A,00,00,00,A9,90,0560
29 DATA 8D,E5,C7,A2,05,A0,00,84,1028
30 DATA FD,A9,C0,85,FE,A9,08,85,1311
31 DATA FC,A9,78,85,FB,B1,FB,91,1498
32 DATA FD,C8,D0,F9,E6,FC,E6,FE,1876
33 DATA CA,D0,F2,A0,48,B1,FB,91,1457
34 DATA FD,88,10,F9,4C,00,C0,78,1042
35 DATA A2,08,BD,0E,C0,9D,11,03,0742
36 DATA CA,10,F7,30,09,48,B2,00,0772
37 DATA 31,EA,66,FE,00,C0,58,A9,1088
38 DATA 00,8D,20,D0,8D,21,D0,A5,0928
39 DATA 0D,8D,86,02,20,44,E5,20,0651
40 DATA AA,C3,29,20,20,2A,2A,2A,0596
41 DATA 2A,2A,20,20,53,59,53,54,0487
42 DATA 45,4D,55,48,52,2D,36,34,0536
43 DATA 20,42,59,20,41,2E,54,2E,0460
44 DATA 48,2E,20,20,2A,2A,2A,2A,0350
45 DATA 2A,0D,20,20,A9,A3,A2,24,0649
46 DATA 20,D2,FF,CA,D0,FA,20,AA,1359
47 DATA C3,19,0D,0D,0D,20,55,48,0448
48 DATA 52,5A,45,49,54,20,28,48,0542
49 DATA 48,3A,4D,4D,3A,53,53,29,0549
50 DATA 20,3A,20,20,E4,FF,C9,30,0886
51 DATA 30,F9,C9,33,10,F5,20,D2,1052
52 DATA FF,38,E9,30,0A,0A,0A,0A,0632
53 DATA 85,02,20,E4,FF,C9,30,30,0947
54 DATA F9,C9,3A,10,F5,C9,34,30,1070
55 DATA 06,A6,02,E0,20,F0,EB,20,0937
56 DATA D2,FF,38,E9,30,18,65,02,0929
57 DATA AA,AD,0E,DD,09,80,8D,0E,0870
58 DATA DD,AD,0F,DD,29,7F,8D,0F,0954
59 DATA DD,8A,C9,12,30,05,09,80,0768
60 DATA 38,E9,12,20,13,C5,A9,02,0726
61 DATA 85,FD,A9,3A,20,D2,FF,20,1142
62 DATA E4,FF,C9,30,30,F9,C9,36,1284
63 DATA 10,F5,20,D2,FF,38,E9,30,1095
64 DATA 0A,0A,0A,0A,85,02,20,E4,0435
65 DATA FF,C9,30,30,F9,C9,3A,10,1076
66 DATA F5,20,D2,FF,38,E9,30,18,1103
67 DATA 65,02,A6,FD,9D,08,DD,C6,1106
68 DATA FD,D0,C7,A9,00,8D,08,DD,1199
69 DATA 20,AA,C3,19,0D,0D,0D,20,0493
70 DATA 44,41,54,55,4D,20,20,28,0483
71 DATA 4A,4A,3A,4D,4D,3A,54,54,0586
```

SYS 52531 utasítással újra aktiválható. Nem várható, hogy valamely más programmal együtt alkalmazva probléma, ütközés merülne fel. Ha mégis, akkor a színes listázó lekapcsolható a következő két utasítással, amiket direkt módban így kettősponttal elválasztva kell ebben a sorrendben beadni:

SYS 65418:SYS 58451

A „színes listázó" működése a legfontosabb rendszervektorok elhajlításán alapszik. A LIST-vektor a program egy rutinjára mutat, ami a tokeneket és a REM-sorokat felismeri és ennek megfelelően színezi. A kimenet-vektor felismeri a sorvégeket és ha szükséges, lekapcsolja a REM-színezést. Az utasításvektor elhajlítása révén a LIST-parancs beadásakor bekapcsolódik az első jelzőszín, majd felismeri az új felkiáltójel-parancsot. A melegstart-vektor új rutinra mutat, ami az utasítás végrehajtása után beállítja a direkt-mód színét. A direkt-mód vektora el van hajlítva, és a direkt-mód színeinek a beállítása mellett arról is gondoskodik, hogy a kimenet vektora a ⟨RUN/STOP-RESTORE⟩-ral való kiszállás után az új értékre legyen elhajlítva. Ezáltal a „színes listázó" immunná, érzéketlenné válik ezen a billentyűkombinációval szemben.

```
72 DATA 29,20,20,3A,20,20,E4,FF,0710
73 DATA C9,30,30,F9,C9,3A,10,F5,1066
74 DATA 20,D2,FF,38,E9,30,0A,0A,0854
75 DATA 0A,0A,85,02,20,E4,FF,C9,0871
76 DATA 30,30,F9,C9,3A,10,F5,20,0897
77 DATA D2,FF,38,E9,30,18,65,02,0929
78 DATA 20,DC,C3,8D,F4,C7,A9,3A,1258
79 DATA 20,D2,FF,20,E4,FF,C9,30,1261
80 DATA 30,F9,C9,32,10,F5,20,D2,1051
81 DATA FF,38,E9,30,0A,0A,0A,0A,0632
82 DATA 85,02,20,E4,FF,C9,30,30,0947
83 DATA F9,C9,3A,10,F5,C9,33,30,1069
84 DATA 06,A6,02,E0,10,F0,EB,20,0921
85 DATA D2,FF,38,E9,30,18,65,02,0929
86 DATA 20,DC,C3,8D,F3,C7,A9,3A,1257
87 DATA 20,D2,FF,20,E4,FF,C9,30,1261
88 DATA 30,F9,C9,34,10,F5,20,D2,1053
89 DATA FF,38,E9,30,0A,0A,0A,0A,0632
90 DATA 85,02,20,E4,FF,C9,30,30,0947
91 DATA F9,C9,3A,10,F5,20,D2,FF,1266
92 DATA 38,E9,30,18,65,02,20,DC,0716
93 DATA C3,D0,0C,1F,1C,1F,1E,1F,0566
94 DATA 1E,1F,1F,1E,1F,1E,1F,AE,0388
95 DATA F3,C7,85,02,CA,BD,D4,C1,1373
96 DATA C5,02,10,06,20,FD,C3,4C,0777
97 DATA 11,C1,A5,02,8D,F2,C7,D0,1167
98 DATA 4C,05,05,06,06,06,06,06,0116
99 DATA C4,EC,14,3C,64,8C,B4,06,0938
100 DATA 08,08,0A,07,09,07,0D,0F,0077
101 DATA 0E,14,01,07,04,09,05,0E,0074
102 DATA 13,14,01,07,0D,09,14,14,0109
103 DATA 17,0F,03,08,04,0F,0E,0E,0096
104 DATA 05,12,13,14,01,07,06,12,0094
105 DATA 05,09,14,01,07,13,0F,0E,0090
106 DATA 0E,01,02,05,0E,04,13,0F,0074
107 DATA 0E,0E,14,01,07,20,0B,C4,0295
108 DATA A2,00,86,FE,86,02,BD,FA,1125
109 DATA C1,85,FC,BD,01,C2,85,FB,1346
110 DATA BD,08,C2,85,FD,86,FE,A6,1331
111 DATA 02,A0,00,BD,0F,C2,91,FB,0956
112 DATA E8,C8,C4,FD,D0,F5,86,02,1470
113 DATA A6,FE,E8,E0,07,D0,D7,A9,1475
114 DATA 00,85,02,20,22,C4,20,E4,0657
115 DATA FF,C9,91,F0,20,C9,11,F0,1331
116 DATA 07,C9,0D,D0,F1,4C,B3,C2,1119
117 DATA 20,22,C4,E6,02,A5,02,C9,0862
118 DATA 07,D0,04,A9,00,85,02,20,0555
119 DATA 22,C4,4C,7F,C2,20,22,C4,0889
120 DATA C6,02,10,F3,A9,06,85,02,0769
121 DATA D0,ED,A6,02,8E,F5,C7,20,1231
122 DATA A9,C4,A2,03,BD,08,DD,9D,1105
123 DATA EE,C7,CA,10,F7,AD,F1,C7,1515
124 DATA 8D,EE,C7,20,F1,C4,8D,F1,1429
125 DATA C7,78,A9,00,8D,11,03,8D,0790
126 DATA 16,03,A9,C0,8D,12,03,8D,0689
127 DATA 17,03,A9,F4,8D,14,03,A9,0772
128 DATA C2,8D,15,03,58,20,44,E5,0776
129 DATA 4C,1D,C5,A2,03,BD,08,DD,0885
130 DATA 9D,EB,C7,CA,10,F7,AD,0B,1240
131 DATA DD,C9,80,D0,03,20,2D,C5,1035
132 DATA AD,EC,C7,CD,EF,C7,D0,03,1462
133 DATA 4C,31,EA,A2,02,78,20,F1,0916
134 DATA C4,8D,EE,C7,BD,EC,C7,9D,1555
135 DATA EF,C7,CA,10,F7,E8,BD,EF,1563
136 DATA C7,D0,08,E8,E0,03,D0,F6,1328
137 DATA 4C,3C,C4,A2,27,A9,20,9D,0891
138 DATA 00,04,CA,10,FA,AE,F5,C7,1090
139 DATA BD,08,C2,AA,BD,F5,C7,9D,1351
140 DATA 02,04,CA,D0,F7,AD,F2,C7,1277
141 DATA 20,95,C4,8E,11,04,8C,12,0698
142 DATA 04,AD,F3,C7,20,95,C4,8E,1138
143 DATA 14,04,8C,15,04,AD,F4,C7,0805
144 DATA 20,95,C4,8E,17,04,8C,18,0710
145 DATA 04,AD,F1,C7,20,80,C4,8E,1115
146 DATA 1D,04,8C,1E,04,AD,F0,C7,0819
147 DATA 20,80,C4,8E,20,04,8C,21,0707
148 DATA 04,AD,EF,C7,20,80,C4,8E,1113
149 DATA 23,04,8C,24,04,A9,2E,8D,0575
150 DATA 13,04,8D,16,04,A9,3A,8D,0558
151 DATA 1F,04,8D,22,04,58,4C,7E,0504
152 DATA EA,78,68,85,FB,68,85,FC,1331
153 DATA A9,0D,8D,86,02,A0,01,B1,0797
154 DATA FB,85,02,E6,02,C8,E6,02,1050
155 DATA B1,FB,20,D2,FF,C8,C4,02,1323
156 DATA D0,F6,C6,02,18,A5,FB,65,1195
157 DATA 02,AA,A5,FC,69,00,48,8A,0904
158 DATA 48,58,60,85,FB,A9,00,85,0942
159 DATA FC,A5,FB,4A,4A,4A,4A,AA,1134
160 DATA F0,0A,A5,FC,18,69,0A,85,0939
161 DATA FC,CA,D0,F6,A5,FB,29,0F,1380
162 DATA 18,65,FC,60,A2,07,20,FF,0929
163 DATA E9,18,A2,04,A0,27,20,0A,0664
164 DATA E5,60,A2,03,A9,D9,85,FC,1261
165 DATA A0,00,84,FB,A9,0D,91,FB,1121
166 DATA C8,D0,FB,E6,FC,CA,D0,F6,1797
167 DATA 60,A6,02,BD,FA,C1,85,FC,1281
168 DATA BD,01,C2,85,FB,BC,08,C2,1158
169 DATA 88,B1,FB,49,80,91,FB,88,1297
170 DATA 10,F7,60,AE,F5,C7,E8,E0,1433
171 DATA 07,D0,02,A2,00,8E,F5,C7,0965
172 DATA 20,A9,C4,AE,F3,C7,EE,F2,1493
173 DATA C7,AD,F2,C7,CA,DD,D4,C1,1641
174 DATA 30,22,F0,20,A9,01,8D,F2,0907
175 DATA C7,E8,E8,8E,F3,C7,E0,0D,1484
176 DATA D0,12,A2,01,8E,F3,C7,AE,1147
177 DATA F4,C7,E8,E0,64,D0,02,A2,1371
178 DATA 00,8E,F4,C7,4C,7E,EA,8D,1162
179 DATA EA,C7,4A,4A,4A,4A,18,69,0858
180 DATA 30,AA,AD,EA,C7,29,0F,18,0904
181 DATA 69,30,A8,60,A2,00,38,E9,0868
182 DATA 0A,30,03,E8,D0,F9,18,69,0879
183 DATA 3A,A8,8A,18,69,30,AA,60,0807
184 DATA AE,F5,C7,BD,08,C2,A8,88,1313
185 DATA E0,00,F0,32,E0,01,F0,29,1020
186 DATA E0,02,F0,20,E0,03,F0,17,0988
187 DATA E0,04,F0,0E,E0,05,F0,05,0956
188 DATA B9,3F,C2,D0,1C,B9,36,C2,1111
189 DATA D0,17,B9,2F,C2,D0,12,B9,1068
190 DATA 25,C2,D0,0D,B9,1D,C2,D0,1068
191 DATA 08,B9,15,C2,D0,03,B9,0F,0819
192 DATA C2,99,F6,C7,88,10,C1,60,1233
193 DATA AD,EE,C7,10,15,29,7F,F8,1063
194 DATA 18,69,12,D8,C9,24,F0,06,0846
195 DATA 30,07,38,E9,06,60,38,E9,0735
196 DATA 12,60,C9,12,D0,FB,38,E9,1081
197 DATA 12,60,C9,80,D0,02,A9,12,0840
198 DATA 8D,0B,DD,60,AD,E5,C7,F0,1310
199 DATA 08,A9,00,8D,E5,C7,4C,7B,0945
200 DATA E3,4C,74,A4,A2,03,BD,08,0945
201 DATA DD,9D,E6,C7,CA,10,F7,A9,1441
202 DATA 90,8D,E9,C7,A2,03,BD,E6,1301
203 DATA C7,9D,08,DD,CA,10,F7,60,1146
204 DATA 34,44,0120

READY.
```

Forrásprogramkészítő

A gépi kódban a programozók gyakran „vesznek kölcsön" rutinokat a ROM-ból. Ezeket azután módosítva, vagy csupán transzponálva illesztik be saját „opus"-ukba. Ez általában úgy történik hogy a ROM-listából, a forrásprogramba „pötyögik be" a felhasználni kívánt rutint. Ezt a meglehetősen gépies, és tévedésekre módot adó tevékenységet „váltja ki" az ismertetésre kerülő program.

A PROGRAM HASZNÁLATA: A behívást és indítást követően, a kiírandó rutin kezdő-, és végcímének, valamint az eltolási címnek a beírása után indul az „érdemi munka". A program egyaránt elfogad decimális és hexadecimális számot, de csak a szimbólumával (#;$) együtt. A tulajdonképpeni forrásprogram 1000-es sorszámmal kezdődik, és a sorszámok 10-esével növekednek. (Lásd 4. listától kezdve.)

Transzportált forrásprogram iratása esetén, csak a kijelölt intervallumba eső címeket írja át a program (5. lista).

Mindkét üzemmódban, ha a feltételes ugrások címei kívül esnek a megadott kezdő-, ill. végcímen, akkor a program *-gal jelöli ezeket a sorokat.

A végcím elérése után a bekért névvel lemezre menti forrásprogramunkat, majd újraindul a címek bekérésével.

A PROGRAM MŰKÖDÉSE: Indítás után, a $C000 címtől kezdődően a memóriába töltődnek a program működéséhez szükséges adatok. A címek beírása után a program kiolvassa az első byte-ot, amely az első utasítás kódja. E kód alapján, a $C000 címen kezdődő területen keresi meg az utasításkódhoz tartozó mnemonikot, az utasítás hosszát, valamit a változót, amely a helyes szintaxis kialakításához szükséges befejező karaktereket [I$(0)—I$(5)] jelöli ki. Ennek utána az utasítás hosszától függően további byte-ot vagy byte-okat (alsó, ill. felső byte) olvas ki. Kiszámítja decimális értéküket és hexadecimális számmá konvertálja. Ezután „összeszereli" a sort: sorszám+mnemonik+hexadecimális cím. (Ha szükséges, kiegészítő karaktereket és/vagy *-ot helyez el.)

Végül beállítja a következő utasítás kezdőcímét (ac), a sorszámot, majd a DATAMAKER-ekhez hasonlóan a módosított data sort, és a kreált sort beírja önmagába. A végcímet elérve, az utolsó sor „XXXX .END" lesz.

A befejező rutin (890—990) kiszámítja az 1000. sor térbeli kezdőcímét, majd ide állítja a 43/44-es mutatót (BASIC kezdet). Ehhez a számításhoz szolgál „jelző cölöpként" a 990-es datasor. A lemezrementés befejeztével, a mutatókat (43/44—45/46) úgy állítjuk át, hogy az interpreter csak a főprogramot „lássa".

Befejezésül néhány kommentár a mellékelt listákhoz:

1. lista maga a főprogram. Begépelésekor ügyeljünk a 220; 230; 460 és 480-as sorok „szöveghűségére", valamint arra, hogy a sorszámozás is megegyezzen!! De a 990-es sorról sem feledkezhetünk meg!

1/a lista a befejező rutin egy másik lehetséges változata. Míg az eredeti megoldás a „következő data elem" mutatót (65/66) használja az 1000. sor lokalizálására, ez a megoldás a programmutatót (122/123). Továbbá, nem használja a SAVE utasítást, hanem a megnyitott file-ba másolja a forrásprogramot.

2. lista a „MNEMONIK" nevű file BASIC betöltője. Ha a program a datasorokban hibát talál, az alábbi sorokat írja ki:

ADATHIBA!!!
A HIBÁS SOR.
c=xxxx:sys 866:goto 30

A kurzor a hibás sor elején villog.

A hiba kijavítása után: ⟨RETURN⟩. A kurzor a "c=....." sorra kerül. Újabb ⟨RETURN⟩ után, a program a javított sor újrafeldolgozásával folytatódik. Végül a tárba töltött adatok, a létrehozott „MNEMONIK" nevű file-ba kerülnek. (Ha a hibás sor száma = X, akkor a „RESTORE X" utasítást, a szalagpufferbe töltött gépi kódú rutin emulálja.) Természetesen a „programfej" általános betöltőként is használható! Ez esetben azonban, a 30., a 150. és a 180. sorokat aktualizálni kell.

3. lista a demo célra kiválasztott ROM részletet tartalmazza, a $AD9E—$ADDB címek között. (A lista BASIC nyelven írott MONITOR-programmal készült.)

4. lista „kimásolt" részlete a FREMEVL ROM-rutinnak.

5. lista ugyanez a részlet, de „transzponálva" a $C000 címre.

6. lista a HELP+ által lefordított 5. lista.

Jó munkát!

Szász Sándor

*Felhasznált irodalom: Dr. Úry László COMMODORE 64 II. köt.
L. Englisch, Gépi kódú programozás a ..*
H. T. Weltner, További tippek és ....*
Angerhausen & Tsai., A Commodore 64-....*
A *-gal jelölt művek a DATA BECKER-NOVOTRADE kiadványai.

```
0 rem 1. lista
5 print "■■■■■■ loading mnemonik ... "
10 if q=0 then q=1:load"mnemonik",8,1
20 poke 53281,8:print "■■■chr$(14)
30 print "    ■ Memoriabol forrasprogram. ■■
40 print "    ■ Startcim ■ :";
50 q=6:gosub 560:ac=l-1:fk=l:z=l
60 if e then e=0:goto 40
70 if w=35 then ck$=a$:goto 90
80 ck$=l$
90 print "■  ■ Vegcim   ■ :";
100 q=7:gosub 570:fv=l
110 if e then e=0:gosub 870:goto 90
120 if fv<=fk then gosub 840:goto 20
130 print "■  ■ Eltolasi cim   ■ :";
140 q=9:gosub 570
150 if e then e=0:gosub 870:goto 130
160 if l then ck$=l$:cd=(fk-l)*-1:z=l
170 print "■■   ?? O.K. ??   ■i■/■n■
180 get w$:if w$<>"i" and w$<>"n" then 180
190 if w$="n" then q=6:gosub 870:goto 40
200 v$="*=$"+ck$+"       ; #"+mid$(str$(z),2,5)
210 en=1:s=1000:print chr$(142):goto 460
220 data 0, 0, 0, 0, 0, 0
230 read fk,fv,cd,ac,s,en
240 if en=0 then 890
250 q(1)=4096:q(2)=256:q(3)=16:q(4)=1
260 i$(0)="":i$(1)=",x)":i$(2)="),y"
270 i$(3)=",x":i$(4)=",y":i$(5)=")"
280 if ac>fv then en=0
290 d=peek(ac):z=ac+2
300 v$="":for k=0 to 4:c=49152+(d*7)+k
310 v$=v$+chr$(peek(c)):next k
320 c=c+1:h=peek(c):c=c+1:i=peek(c)
330 r=ac:gosub 810
340 if h<2 then 460
350 ac=ac+1:ab=peek(ac):if h=2 then 370
360 ac=ac+1:fb=peek(ac)
370 l=fb*256+ab:if h=3 then 430
380 if asc(v$)<>66 or d=36 then 450
390 if ab>128 then l=z-256+ab:goto 410
```

```
400 l=z+ab
410 if l<fk or l>fv then ex$="  *"
420 goto 440
430 if l<fk or l>fv then 450
440 l=l+cd:r=l:gosub 810
450 gosub 750:v$=v$+"$"+a$
460 restore:print"█goto230"
470 gosub 540
480 print"██220 data"fk"█,"fv"█,";
490 print cd"█,"ac+1"█,"s+10"█,"en
500 a$=str$(s)
510 if en=0 then a$=a$+" .end":goto 530
520 a$=a$+v$+i$(i)+ex$
530 print a$:print"█:end
540 poke198,4:poke631,13:poke632,13
550 poke633,19:poke634,13:return
560 q(1)=4096:q(2)=256:q(3)=16:q(4)=1
570 poke 19,1:input l$:poke 19,0
580 w=asc(l$):l$=mid$(l$,2,5)
590 print:on w-34 goto 610,660
600 e=1:goto 840
610 l=val(l$):if q=9 then 630
620 if l<=0 then 650
630 r=l:gosub 810
640 goto 730
650 e=1:goto 720
660 l$=right$("000"+l$,4)
670 l=0:for i=1 to 4:x=asc(mid$(l$,i,1))
680 if x>47 and x<58 then b=48:goto 710
690 if x>64 and x<71 then b=55:goto 710
700 i=4:e=1:goto 720
710 l=l+(x-b)*q(i)
720 next i:if e then l=0:goto 840
730 if w=35 then 750
740 goto 800
750 n=3:if l>255 then n=1
760 a=l:a$="":for j=n to 4
770 b=int(a/q(j)):a=a-(b*q(j))
780 y=55:if b<10 then y=48
790 a$=a$+chr$(b+y):next j
800 return
810 if r+h<=65535 then return
820 print"███  █ owerflow █ <f-1>"
830 e=2:goto 850
840 print "██  █ cimhiba !!  <f-1> █ "
850 get w$:if w$<>chr$(133) then 850
860 if e=2 then run 20
870 for i=q to 20:poke 781,i:sys 59903
880 next i:poke 214,q:sys 58640:return
890 for l=1 to 7:read a:x=peek(65)+1
900 print"████ a forrasprogram neve:";
910 input fi$
920 poke 43,x:poke 44,17
930 save fi$,8
940 poke 43,1:poke 44,8
950 ec=4352+x
960 poke ec,0:poke ec+1,0
970 poke 45,x+2:poke 46,17
980 run 20
990 data 0

ready.
```

```
100 rem 1/a lista
270 if en=0 then 990
300 :
890 poke 2,peek(4352+ab):kc=4352+peek(2)
900 print"████ a forrasprogram neve:";
910 input fi$
920 open2,8,2,fi$+",p,w"
930 print#2,chr$(1);chr$(8);
940 for l=kc to peek(46)*256+peek(45)-1
950 print#2,chr$(peek(l));:next l
960 close2:poke kc,0:poke kc+1,0
970 poke 45,peek(2)+2:poke 46,17
980 run 20
990 ab=peek(122)-13:goto 890

ready.
```

```
 0 rem 2. lista
 5 rem "BASIC betolto program
 10 :
 20 for l=830 to 875:read d:poke l,d:next l:print chr$(14)
 30 ck=49152: rem $c000
 40 read v$:if v$="@" then 110
 50 cc=c:s=peek(63)+peek(64)*256
 60 print "█████:s
 70 read v:cn=0:for l=1 to len(v$) step 2
 80 d=(asc(mid$(v$,l,1))-65)*16+asc(mid$(v$,l+1,1))-65:cn=cn+d
 90 poke(ck+c),d:c=c+1:next l:if v=cn then 40
100 goto 190
110 print "██ █ O.K. O.K. O.K.  ":cv=ck+c-1
120 print "██ A programlemez O.K. ?? █I█gen/█N█em"
130 get w$:if w$<>"i" then 130
140 open15,8,15,"i":x=int(ck/256):y=ck-x*256
150 open2,8,2,"mnemonik,p,w":print#2,chr$(y);chr$(x);:for l=ck to cv
160 print#2,chr$(peek(l));:next l:close2:close15
170 print "██ █ O.K. O.K. O.K. !! █:end
180 rem a file neve: mnemonik
190 print "████ █ A D A T H I B A  !! ██
200 print mid$(str$(s),2,5);"data";:sys 830:print
210 print "c="str$(cc)":sys 866:goto 30███:end
```

```
220 data 166,63,134,20,166,64,134,21
230 data 32,19,166,165,95,164,96,56
240 data 233,1,176,1,136,141,60,3
250 data 140,61,3,24,105,6,144,1
260 data 200,76,30,171,173,60,3,172
270 data 61,3,76,36,168,0
280 :
300 data ecfcelcacaabaaepfcebcaciacabdpca,684
310 data cacacaaaaadpcacacacaaaaadpcacaca,446
320 data caaaaaepfcebcacaacaaebfdemcacaac,614
330 data aadpcacacacaaaaafaeifacacaabaaep,567
340 data fcebcacdacaaebfdemcaebabaadpcaca,665
350 data cacaaaaadpcacacacaaaaaepfcebcaca,545
360 data adaaebfdemcacaadaadpcacacacaaaaa,485
370 data ecfaemcacaacaaepfcebcaciacacdpca,685
380 data cacacaaaaadpcacacacaaaaadpcacaca,446
390 data caaaaaepfcebcacaacadebfdemcacaac,617
400 data addpcacacacaaaaaedemedcacaabaaep,548
410 data fcebcacaadaedpcacacacaaaaadpcaca,536
420 data cacaaaaadpcacacacaaaaaepfcebcaca,545
430 data adadebfdemcacaadaddpcacacacaaaaa,491
440 data ekfdfccacaadaaebeoeecaciacabdpca,687
450 data cacacaaaaadpcacacacaaaaaecejfeca,542
460 data caacaaebeoeecacaacaafcepemcacaac,614
470 data aadpcacacacaaaaafaemfacacaabaaeb,557
480 data eoeecacdacaafcepemcaebabaadpcaca,677
490 data cacaaaaaecejfecacaadaaebeoeecaca,629
500 data adaafcepemcacaadaadpcacacacaaaaa,498
510 data ecenejcacaacaaebeoeecaciacacdpca,664
520 data cacacaaaaadpcacacacaaaaadpcacaca,446
530 data caaaaaebeoeecacaacadfcepemcacaac,615
540 data addpcacacacaaaaafdefedcacaabaaeb,543
550 data eoeecacaadaedpcacacacaaaaadpcaca,535
560 data cacaaaaadpcacacacaaaaaebeoeecaca,530
570 data adadfcepemcacaadaddpcacacacaaaaa,504
580 data fcfeejcacaabaaefepfccaciacabdpca,704
590 data cacacaaaaadpcacacacaaaaadpcacaca,446
600 data caaaaaefepfccacaacaaemfdfccacaac,635
610 data aadpcacacacaaaaafaeiebcacaabaaef,542
620 data epfccacdacaaemfdfccaebabaadpcaca,696
630 data cacaaaaaekenfacacaadaaefepfccaca,656
640 data adaaemfdfccacaadaadpcacacacaaaaa,502
650 data ecfgedcacaacaaefepfccaciacacdpca,686
660 data cacacaaaaadpcacacacaaaaadpcacaca,446
670 data caaaaaefepfccacaacademfdfccacaac,638
680 data addpcacacacaaaaaedemejcacaabaaef,544
690 data epfccacaadaedpcacacacaaaaadpcaca,550
700 data cacaaaaadpcacacacaaaaaefepfccaca,549
710 data adademfdfccacaadaddpcacacacaaaaa,508
720 data fcfefdcacaabaaebeeedcaciacabdpca,684
730 data cacacaaaaadpcacacacaaaaadpcacaca,446
740 data caaaaaebeeedcacaacaafcepfccacaac,607
750 data aadpcacacacaaaaafaemebcacaabaaeb,542
760 data eeedcacdacaafcepfccaebabaadpcaca,672
770 data cacaaaaaekenfacaciadafebeeedcaca,639
780 data adaafcepfccacaadaadpcacacacaaaaa,504
790 data ecfgfdcacaacaaebeeedcaciacacdpca,672
800 data cacacaaaaadpcacacacaaaaadpcacaca,446
810 data caaaaaebeeedcacaacadfcepfccacaac,610
820 data addpcacacacaaaaafdefejcacaabaaeb,549
830 data eeedcacaadaedpcacacacaaaaadpcaca,524
840 data cacaaaaadpcacacacaaaaaebeeedcaca,519
850 data adadfcepfccacaadaddpcacacacaaaaa,510
```

3. LISTA

```
AD9E  A6 7A     LDX  $7A
ADA0  D0 02     BNE  $ADA4
ADA2  C6 7B     DEC  $7B
ADA4  C6 7A     DEC  $7A
ADA6  A2 00     LDX  #$00
ADA8  24 48     BIT  $48
ADAA  8A        TXA
ADAB  48        PHA
ADAC  A9 01     LDA  #$01
ADAE  20 FB A3  JSR  $A3FB
ADB1  20 83 AE  JSR  $AE83
ADB4  A9 00     LDA  #$00
ADB6  85 4D     STA  $4D
ADB8  20 79 00  JSR  $0079
ADBB  38        SEC
ADBC  E9 B1     SBC  #$B1
ADBE  90 17     BCC  $ADD7
ADC0  C9 03     CMP  #$03
ADC2  B0 13     BCS  $ADD7
ADC4  C9 01     CMP  #$01
ADC6  2A        ROL  A
ADC7  49 01     EOR  #$01
ADC9  45 4D     EOR  $4D
ADCB  C5 4D     CMP  $4D
ADCD  90 61     BCC  $AE30
ADCF  85 4D     STA  $4D
ADD1  20 73 00  JSR  $0073
ADD4  4C BB AD  JMP  $ADBB
ADD7  A6 4D     LDX  $4D
ADD9  D0 2C     BNE  $AE07
ADDB  B0 7B     BCS  $AE58
ADDD  69 .END
```

```
860 data dpcacacacaaaaafdfeebcaciacabdpca,593
870 data cacacaaaaadpcacacacaaaaafdfefjca,575
880 data caacaafdfeebcacaacaafdfeficacaac,653
890 data aadpcacacacaaaaaeeeffjcacaabaadp,545
900 data cacacacaaaaafefiebcacaabaadpcaca,557
910 data cacaaaaafdfefjcacaadaafdfeebcaca,683
920 data adaafdfeficacaadaadpcacacacaaaaa,516
930 data ecededcacaacaafdfeebcaciacacdpca,669
940 data cacacaaaaadpcacacacaaaaafdfefjca,575
950 data caacadfdfeebcacaacadfdfeficacaac,659
960 data addpcacacacaaaaafefjebcacaabaafd,580
970 data feebcacaadaefefifdcacaabaadpcaca,667
980 data cacaaaaadpcacacacaaaaafdfeebcaca,551
990 data adaddpcacacacaaaaadpcacacacaaaaa,388
1000 data emeefjcacdacaaemeeebcaciacabemee,730
1010 data ficacdacaadpcacacacaaaaaemeefjca,613
1020 data caacaaemeeebcacaacaaemeeficacaac,607
1030 data aadpcacacacaaaaafeebfjcacaabaaem,570
1040 data eeebcacdacaafeebficacaabaadpcaca,631
1050 data cacaaaaaemeefjcacaadaaemeeebcaca,637
1060 data adaaemeeficacaadaadpcacacacaaaaa,493
1070 data ecedfdcacaacaaemeeebcaciacacdpca,662
1080 data cacacaaaaadpcacacacaaaaaemeefjca,552
1090 data caacademeeebcacaacademeeficacaac,613
1100 data aedpcacacacaaaaaedemfgcacaabaaem,565
1110 data eeebcacaadaefefdficacaabaadpcaca,651
1120 data cacaaaaaemeefjcacaadademeeebcaca,640
1130 data adademeeficacaadaedpcacacacaaaaa,500
1140 data edfafjcacdacaaedenfacaciacabdpca,699
1150 data cacacaaaaadpcacacacaaaaaedfafjca,555
1160 data caacaaedenfacacaacaaeeefedcacaac,594
1170 data aadpcacacacaaaaaejeofjcacaabaaed,563
1180 data enfacacdacaaeeeffficacaabaadpcaca,643
1190 data cacaaaaaedfafjcacaadaaedenfacaca,655
1200 data adaaeeefedcacaadaadpcacacacaaaaa,465
1210 data eceoefcacaacaaedenfacaciacacdpca,674
1220 data cacacaaaaadpcacacacaaaaadpcacaca,446
1230 data caaaaaedenfacacaacadeeefedcacaac,595
1240 data addpcacacacaaaaaedemeecacaabaaed,537
1250 data enfacacaadaedpcacacacaaaaadpcaca,546
1260 data cacaaaaadpcacacacaaaaaedenfacaca,543
1270 data adadeeefedcacaadaddpcacacacaaaaa,471
1280 data edfaficacdacaafdecedcaciacabdpca,690
1290 data cacacaaaaadpcacacacaaaaaedfafica,554
1300 data caacaafdecedcacaacaaejeoedcacaac,600
1310 data aadpcacacacaaaaaejeoficacaabaafd,578
1320 data ecedcacdacaaeoepfacacaabaadpcaca,631
1330 data cacaaaaaedfaficacaadaafdecedcaca,646
1340 data adaaejeoedcacaadaadpcacacacaaaaa,479
1350 data eceffbcacaacaafdecedcaciacacdpca,669
1360 data cacacaaaaadpcacacacaaaaadpcacaca,446
1370 data caaaaafdecedcacaacaeejeoedcacaac,602
1380 data addpcacacacaaaaafdefeecacaabaafd,562
1390 data ecedcacaadaedpcacacacaaaaadpcaca,522
1400 data cacaaaaadpcacacacaaaaafdecedcaca,535
1410 data adadejeoedcacaadaddpcacacacaaaaa,485
1420 data aaaaaa,0,@

ready.
```

```
100 ; 4. LISTA
500 ;
1000 *=$AD9E   ; #44446
1010 LDX  $7A
1020 BNE  $ADA4
1030 DEC  $7B
1040 DEC  $7A
1050 LDX #$00
1060 BIT  $48
1070 TXA
1080 PHA
1090 LDA #$01
1100 JSR  $A3FB
1110 JSR  $AE83
1120 LDA #$00
1130 STA  $4D
1140 JSR  $79
1150 SEC
1160 SBC #$B1
1170 BCC  $ADD7
1180 CMP #$03
1190 BCS  $ADD7
1200 CMP #$01
1210 ROL A
1220 EOR #$01
1230 EOR  $4D
1240 CMP  $4D
1250 BCC  $AE30  *
1260 STA  $4D
1270 JSR  $73
1280 JMP  $ADBB
1290 LDX  $4D
1300 BNE  $AE07  *
1310 BCS  $AE58  *
1320 .END

READY.
```

```
FORRAS

1000          100       ;  6. LISTA
1000          500       ;
C000         1000       *=$C000      ; #49152
C000 A6 7A   1010       LDX  $7A
C002 D0 02   1020       BNE  $C006
C004 C6 7B   1030       DEC  $7B
C006 C6 7A   1040       DEC  $7A
C008 A2 00   1050       LDX  #$00
C00A 24 48   1060       BIT  $48
C00C 8A      1070       TXA
C00D 48      1080       PHA
C00E A9 01   1090       LDA  #$01
C010 20 FB A3 1100      JSR  $A3FB
C013 20 83 AE 1110      JSR  $AE83
C016 A9 00   1120       LDA  #$00
C018 85 4D   1130       STA  $4D
C01A 20 79 00 1140      JSR  $79
C01D 38      1150       SEC
C01E E9 B1   1160       SBC  #$B1
C020 90 17   1170       BCC  $C039
C022 C9 03   1180       CMP  #$03
C024 B0 13   1190       BCS  $C039
C026 C9 01   1200       CMP  #$01
C028 2A      1210       ROL  A
C029 49 01   1220       EOR  #$01
C02B 45 4D   1230       EOR  $4D
C02D C5 4D   1240       CMP  $4D
C02F 90 61   1250       BCC  $C092  *
C031 85 4D   1260       STA  $4D
C033 20 73 00 1270      JSR  $73
C036 4C 1D C0 1280      JMP  $C01D
C039 A6 4D   1290       LDX  $4D
C03B D0 2C   1300       BNE  $C069  *
C03D B0 7B   1310       BCS  $C0BA  *
C03F         1320       .END

 LINES:35   SYMBOLE:0   ERRORS:0
```

```
100 ;  5. LISTA
500 ;
1000 *=$C000       ; #49152
1010 LDX  $7A
1020 BNE  $C006
1030 DEC  $7B
1040 DEC  $7A
1050 LDX  #$00
1060 BIT  $48
1070 TXA
1080 PHA
1090 LDA  #$01
1100 JSR  $A3FB
1110 JSR  $AE83
1120 LDA  #$00
1130 STA  $4D
1140 JSR  $79
1150 SEC
1160 SBC  #$B1
1170 BCC  $C039
1180 CMP  #$03
1190 BCS  $C039
1200 CMP  #$01
1210 ROL  A
1220 EOR  #$01
1230 EOR  $4D
1240 CMP  $4D
1250 BCC  $C092  *
1260 STA  $4D
1270 JSR  $73
1280 JMP  $C01D
1290 LDX  $4D
1300 BNE  $C069  *
1310 BCS  $C0BA  *
1320 .END

READY.
```

Tippek Trükkök

A Mozgó Háttér nevű rutin látványos hátteret rajzol, amit saját programjainkban is használhatunk.

```
100 REM ******************************
110 REM *   MOZGO HATTER        C-64 *
120 REM *       ,  ,                 *
130 REM *    KESZITETTE: GABRYEL     *
140 REM ******************************
150 :
160 FOR I=49152 TO 49319
170 READ A:POKE I,A:S=S+A
180 NEXT
190 IF S<>16885 THEN PRINT"HIBA"
200 DATA 120,165,001,072,041,251,133
201 DATA 001,169,000,133,251,169,208
202 DATA 133,252,169,000,133,253,169
203 DATA 048,133,254,160,000,162,000
204 DATA 177,251,145,253,200,208,249
205 DATA 232,230,252,230,254,224,016
206 DATA 208,240,104,133,001,169,028
207 DATA 141,024,208,088,120,169,000
208 DATA 133,002,169,070,141,020,003
209 DATA 169,192,141,021,003,088,096
210 DATA 165,002,024,105,008,133,002
211 DATA 201,064,208,004,169,000,133
212 DATA 002,166,002,160,000,189,104
213 DATA 192,153,000,049,200,232,192
214 DATA 008,208,244,076,049,234
215 DATA 255,001,001,001,001,001,01,1
216 DATA 002,255,002,002,002,002,002,2
217 DATA 004,004,255,004,004,004,004,4
218 DATA 008,008,008,255,008,008,008,8
219 DATA 016,016,016,016,255,016,016,16
220 DATA 032,032,032,032,32,255,032,32
221 DATA 064,064,064,064,064,64,255,64
222 DATA128,128,128,128,128,128,128,255
223 SYS 49152
224 END
```

Alkalmazói program pályázat

Grafikus (Mini) iratkészítő

Felhasználói leírás

Géptípus: Commodore Plus/4 vagy bővített C16.
Nyelv: Gépi kódú és egyéb bővítés nélküli Basic V3.5.

Az énáltalam készített iratkészítő egy olyan program, amely a grafikus képernyőt használja a szövegek és grafikák megjelenítésére. Az elkészített szöveg tetszés szerint keverhető pontonként megszerkesztett finomfelbontású ábrákkal, rajzokkal. Az így elkészített dokumentum (levél, stb.) tárolható lemezen, ill. kazettán, vagy pedig kinyomtatható.

A program lehetőségei, különlegességei:

— A TEDPAINT című rejzolóprogramban elkészített képek felhasználhatóak a programban, ahol egyszerűen átszerkeszthetők.
— A program a megszerkesztett képeket kétszeres méretben nyomtatja ki.
— Ékezetes karakterkészlet.
— Alakzatok rajzolását megkönnyítő funkció.
— Lehetőség van más karakterkészlet betöltésére is a LOAD rutin segítségével.

FIGYELEM

A program a MPS801/803 típusú nyomtatókkal működik együtt.

A program kezelését az alábbiakban ismertetem.

Indítás után a világoskék alapon, sötét keretben a főmenüt láthatjuk. Itt választhatjuk ki a számunkra megfelelő funkciót a hozzá tartozó számbillentyűk megnyomásával.

A FŐMENÜ

1. Szerkesztés
2. Grafikus szerkesztés
3. Képformátum
4. I/O műveletek
5. Nyomtatvány készítése
6. Kilépés Basic-be

1. Szerkesztés

Ez tulajdonképpen a szövegmód. Ebben az üzemmódban a képernyő mérete maximum 40×25 karakter lehet. Tehát a grafika és szöveg szerkesztéséhez pontosan egy képernyőoldal áll a rendelkezésünkre. A szöveg szerkesztéséhez is a grafikus képernyőt használjuk. A dokumentum tárolása lemezen 33 blokkot foglal le.

1.1. Gépelés

Alapállapotban a kurzor a képernyő bal felső sarkában van. A képernyő mérete alapállásban 38×22 karakter. A program közvetlen betöltése után a grafikus memória mindenféle szemetet tartalmaz. Ezért ha ebbe vagy a grafikus szerkesztő üzemmódba lépünk, a képernyőt le kell törölni a SHIFT+CLEAR/HOME billentyűkkel.

A program minden egyes billentyű lenyomásakor rövid hangjelzést ad. Ez segít abban, hogy gépelés közben ne kelljen mindig a képernyőre figyelnünk.

Ha gépelés közben a sor végére értünk, a kurzor automatikusan a következő sor elejére ugrik. Nem így van ez akkor, ha a kurzor mozgatása közben értünk a sor végére, ilyenkor a kurzor egy helyben áll. Ha az aktuális ablak legalsó sorában gépelünk, akkor a kurzor ha a sor végére ért, visszaugrik az alsó sor elejére.

Ha a kurzor egy karakterre lép, akkor a kurzor felülíródik, és kitörli a karaktert.

1.2. Karakterek

A program magyar ékezetes karaktereket használ. Az ékezetes betűk a következő billentyűkön vannak elhelyezve:

á — SHIFT/+	Á — CBM/+	A „CBM" a Commodore billentyűt jelenti.
é — SHIFT/—	É — CBM/—	
í — SHIFT/@	ó — SHIFT/£	
ü — CBM/@	ű — @	
ö — CBM/£	ő — £	
ú — SHIFT/*		

A CBM/Q, W, E, A, S, D, Z, X, C karaktereket nem használhatjuk.

Lehetőség van inverz karakterek írására is. Az inverz mód a CTRL/9 gombokkal be, a CTRL/0 gombokkal pedig kikapcsolható.

1.3. A szerkesztés folyamán használható billentyűk

CRSR nyilak	— a kurzor mozgatását végzik az aktuális ablakon belül.
RETURN	— a kurzor a következő sor elejére ugrik.
DEL	— a kurzortól balra levő karakterek törlése.
HOME	— a kurzor az aktuális ablak bal felső sarkába kerül.
SHIFT+CLEAR HOME	— a grafikus tár törlése, ablaktól függetlenül.
ESC	— visszalépés a főmenübe.

Ezen kívül még van 9 db olyan billentyű, amelyek a kurzor pozicionálását végzik a munkaképernyőn. A munkaképernyő az éppen aktuális ablakot jelenti.

A billentyűket a C-billentyű nyomva tartása mellett kell használni.

Q, W, E — a kurzor a legfelső sor bal szélére, közepére, ill. jobb szélére ugrik.
A, S, D — ugyanaz mint az előző, csak a középső sorban.
Z, X, C — mint az előzőek, de a legalsó sorban.

2. Grafikus szerkesztés

Ebben az üzemmódban a munkaképernyőre nagyfelbontású ábrákat készíthetünk. A kettes számbillentyű lenyomása után egy almenü jelenik meg a képernyőn:

1. Grafikus kép megtekintése
2. Grafikus tár törlése
3. Paraméterek
4. Szerkesztés
5. Alakzat rajzolása

6. Festés
7. Vissza a főmenübe

2.1. Grafikus kép megtekintése

Megnézhetjük a grafikus képernyő tartalmát az ,,1'' billentyű lenyomására. Bármely más billentyűre visszatérünk az almenübe.

2.2. Grafikus tár törlése

Letörölhetjük a grafikus képernyőt, ha a biztonsági rákérdezésnél ,,I''-t nyomunk.

2.3. Paraméterek

Megváltoztathatjuk a grafikus kurzor paramétereit, amelyek a következők:

1. Grafikus kurzor léptéke:
   Azt határozza meg, hogy a grafikus kurzor hány pixelközönként gyűjtsa ki a pontokat. Értéke 0—64-ig terjedő egész szám lehet.
2. Kurzor ,,X'' szélessége:
   Megadja a grafikus kurzor ,,X'' irányú méretét képpontokban. Értéke 0—32-ig terjedő intervallumba kell hogy essen.
3. Kurzor ,,Y'' szélessége:
   Ugyanaz mint az előbbi csak az ,,Y'' méretre.
4. Kurzor állapota:
   A grafikus kurzor állapotát állíthatjuk be attól függően, hogy rajzolni, vagy radírozni akarunk-e. Ha értéke 0, a kurzor háttérszínű pontokkal, ha 1 akkor kurzorszínű pontokkal jelenik meg.

2.4. Szerkesztés

2.4.1. A rajzolás

A kurzort a CRSR nyilakkal mozgathatjuk. Alapállapotban az éppen aktuális kurzorral, és paraméterekkel rajzolhatunk.

2.4.2. A kurzor mozgatása

A SPACE billentyű lenyomásával áttérhetünk pozícionáló módba. Ilyenkor a CRSR gombokkal nem rajzolunk, hanem a grafikus kurzor bal felső sarkát szimbolizáló pontot mozgathatjuk.

Ha hosszabb utat kell a kurzorral megtenni, akkor használhatjuk a ,,szerkesztés'' üzemmódnál ismertetett pozícionáló billentyűket, a C-gomb nyomva tartása nélkül, továbbá itt létezik még néhány más mozgató gomb is:

, — a kurzor az aktuális sor bal szélére ugrik,
. — a kurzor az aktuális sor jobb szélére ugrik,
; — a kurzor az aktuális oszlop legfelső sorába ugrik,
/ — a kurzor az aktuális oszlop legalsó sorába ugrik.

2.4.3. Egyéb billentyűk

CRSR nyilak — vonal húzása megfelelő irányba.
HOME — a kurzor a bal felső sarokba ugrik.
CLEAR — a grafikus képernyő letörlése.
SPACE — pozícionáló mód.
ESC — visszalépés almenübe.

Pozícionáló üzemmód:

CRSR nyilak — a kurzor mozgatása megfelelő irányba.
HOME — a kurzor az X=10, Y=10 helyzetbe ugrik.
SPACE — rajzoló mód.

2.5. Alakzat rajzolása

Lehetőség van arra, hogy a grafikus képernyőre különböző alakzatokat rajzoljunk. A következő almenüből választhatunk:

1. Kör vagy elipszis
2. Téglalap
3. Egyenes vonal
4. Vissza

2.5.1. Kör vagy elipszis rajzolása

Kör vagy elipszis rajzolásánál a középpont mindig a grafikus kurzor éppen aktuális koordinátája.

A menüpontban megadhatjuk a kör ,,X'' és ,,Y'' sugarát. A kör középpontját úgy állíthatjuk be, hogy a SZERKESZTÉS (4) menüpontban pozicionáljuk a kurzort a megfelelő helyre.

2.5.2. Téglalap rajzolása

A grafikus kurzor helyzete adja a téglalap bal felső sarkát. Nekünk a jobb alsó sarok koordinátáit van módunkban megadnunk.

Meghatározhatjuk, hogy a téglalap hány szöggel legyen elforgatva. Ha zárt téglalapot szeretnénk, akkor még be kell írnunk egy 1-est. Ha bármely más karaktert írunk, akkor normál téglalapot kapunk.

A formátum a következő: JX, JY, szög, kitöltés.

A jobb alsó sarok koordinátájának megadásakor azt kell meghatározni, hogy a jobb alsó sarok hány pixellel legyen távolabb a bal felső saroktól X és Y irányba. Például ha a kurzor X=159, Y=99 pozicióban van, és egy 5×5 pixeles négyzetet szeretnénk rajzolni, akkor a
164,104,0,0 helyett a
5,5,0,0 számsort kell beírni.

2.5.3. Egyenes vonal rajzolása

Itt is a grafikus kurzor adja a vonal egyik végét. Nekünk a másik végpont koordinátáit kell megadnunk. Bármilyen alakzatot rajzolunk, a kurzor helye nem változik megrajzolás után.

2.6. Festés

A képernyőn lévő körbezárt területet be tudjuk festeni. Előtte a kurzort a befestendő területre kell pozicionálni.

3. Képformátum

Beállíthatjuk a munkaképernyő méretét a szövegmódban. Erre akkor lehet szükség, ha a munkaképernyő meghatározott részére akarunk írni anélkül, hogy a kép más részleteit megzavarjuk.

	Alapállapot	Határok
Jobb margó:	38	0—39
Bal margó:	01	0—39
Felső lapszél:	01	0—24
Alsó lapszél:	22	0—24

4. I/O Műveletek

A programmal elkészített dokumentumokat nemcsak nyomtatni tudjuk, hanem tárolhatjuk is lemezen vagy kazettán. Az első lépésben ki kell választani, hogy melyik egységgel kívánunk dolgozni. (K — Kazetta, F — Floppy.)

Ezután négy funkció közül választhatunk:
SAVE LOAD DIRECTORY VISSZA

A nyíl segítségével választhatjuk ki a megfelelőt. A nyilat a CRSR gombokkal mozgathatjuk. A kiválasztott funkciót a SPACE billentyűvel aktiválhatjuk. Kiválasztáskor a nyílnak a funkció második karaktere alatt kell lennie, különben a SPACE hatástalan.

4.1. Save

A munkaképernyő tartalmát tudjuk kimenteni az aktuális egységre. A file neve maximum 13 karakter lehet, és a ,,GR'' kiterjesztéssel kerül kimentésre. A filenév meg-

adása után a gép áttér MONITOR üzemmódba. Az F1 lenyomására indul a mentés.

4.2. LOAD

A már meglévő képfile-t tudjuk betölteni. A töltés ideje alatt a képernyő alján a ,,Töltés'' felirat olvasható.

FIGYELEM: A betöltés elindulásakor a grafikus tár aktuális tartalma elvész.

4.3. Directory

Kiírathatjuk a meghajtóban lévő lemez tartalomjegyzékét.

4.4. Vissza

Visszatérés a főmenübe.

5. Nyomtavány készítése

Kinyomtathatjuk a grafikus képernyő tartalmát MPS801/803 nyomtatóval. Sajnos a nyomtatás sebessége nagyon lassú.

A nyomtatás befejezte előtt helyezzük a lemezt a meghajtóba, vagy a kazettát a magnóba amely a program karakterkészletét tartalmazza. A nyomtatás befejezése után ismét betöltődik a karakterkészlet. Erre azért van szükség, mert a gép a nyomtatás befejeztével általam ismeretlen okból összezavarja a program karakterkészletét.

A program az indítás után meghívja a bejelentkezési felirat kiírását végző szubrutint. Mielőtt ez végrehajtódna, ez az alprogram még továbbugrik a karakterkészletet bekapcsoló sorra, ahol átállítódik az eredeti karakterkészlet a betöltött karakterkészletre. Erre az utóbbi rutinra egyébként sokszor szükség van a program futása során, mivel ha a grafikus képernyőből kilépünk (pl. ESC-kel), akkor ismét be kell kapcsolni a karakterkészletet.

A 110—205. sorokban kiíródik a képernyőre a főmenü. Az ugrásokat egyetlen sor valósítja meg, az ON GOTO utasítással. Ez a megoldás található az összes almenünél is, kivéve az I/O műveleteknél.

A 215—385. sorok a szöveg szerkesztését, helyesebben felírását végzik a grafikus képernyőre. Működési mechanizmusa nagyon egyszerű:

A *K$* szöveges változóba beolvassuk a lenyomott billentyűhöz tartozó karaktert és azt kiírjuk az *X* és *Y* változótól függően a megfelelő oszlopba, ill. sorba. Hogy ne keletkezhessen túl nagy érték az *X* és *Y* változókban, a rutin elején és végén lévő *IF* utasítások a felelősek, melyek összehasonlítják az aktuális sor és oszlopszámot a megengedett, illetve beállított értékkel.

Az I/O műveleteknél a kiválasztott periféria egységszáma eltárolódik a H változóba, ahol későbbi felhasználásra kerül a nyomtatás utáni karakterkészlet töltésnél.

Mint már említettem, az I/O műveletek almenüje máshogy működik mint a többi menü. A nyíl oszlopszámát tároló *O* változó értéke minden jobbra, illetve balra mozgatáskor más-más értéket vesz fel. Az *IF* utasításokkal megvalósított oszloplekérdező rutin ezeket az értékeket figyeli. A lekérdező rutinra a *SPACE* billentyűvel kapcsolunk rá. A rutin egyezteti az oszlopszámot a funkció nevének második karaktere oszlopszámával, és ha egyezik, elugrik a megfelelő sorra, ha eddig nem, kezdődik mindez elölről.

A *SAVE* opció választásakor a filenév bekérése után a program átdefiniálja az *F1*-es funkcióbillentyűt. Az ezt megvalósító programsor azonban hosszabb, mint amenynyit a Basic megenged, ezért a *CHR$* függvényt rövidítve, tehát *CH (C,SHIFT+H)* formában kell bevinni. Ezután a gép áttér *MONITOR* üzemmódra és az *F1*-re indul a mentés.

A rajzolást a következőképpen valósítottam meg:

A rajzolásnál a kurzort lényegében egy *BOX* utasítás rajzolja ki. A mozgató rutinban *X*-hez és *Y*-hoz iránynak megfelelően hozzáadja, illetve kivonja a program a grafikus kurzor léptetéseként beállított értéket, és visszaugrik a *BOX* utasításhoz, ahol a kurzor másik sarkát úgy kapjuk meg, hogy az előbbi sarokponthoz hozzáadjuk a kurzor *X* és *Y* szélességeként megadott értéket. Így a megfelelő nagyságú kurzorral mind a négy irányba rajzolhatunk.

Hibakezelés és hibaelhárítás a programon belül

A program futása során a listában keletkező szintaktikai hibákat figyelmen kívül hagyja, és az azt követő programsoron folytatja a végrehajtást. Ezt az 1295. sorban lévő *RESUME NEXT* utasítás valósítja meg, amely természetesen módosítható.

A program futását a *RUN/STOP* billentyűvel nem lehet megszakítani, mivel az 50. sorban szerepel a *POKE 806, 103* utasítás. A *LOAD* rutin elindulásakor ismét bekapcsolja a program a grafikus memóriát, mivel ha a program betöltése után közvetlen töltünk be egy képfilet, akkor az nem írja felül a program egy részét. Ezért törlődik egy grafika betöltésekor a grafikus memória.

Változólista

H	— az aktuális háttértároló. Ha értéke 1, akkor Datasette, ha 8, 9, 10, 11, akkor meghajtó.
BM	— a bal margó értékét tartalmazza.
JM	— a jobb margó értékét tartalmazza.
FG	— a felső lapszél értékét tárolja.
AR	— az alsó lapszél értékét tárolja.
K$	— a feldolgozásra kerülő karakter.
X	— a grafikus vagy szöveges kurzor éppen aktuális oszlopszáma.
Y	— a grafikus vagy szöveges kurzor éppen aktuális sorszáma.
MD	— szöveges kurzor üzemmódja. Ha értéke 1, akkor inverz, ha 0, akkor normál megjelenítés.
GL	— a grafikus kurzor léptékét tárolja.
XS	— a grafikus kurzor *X* szélességét tárolja.
YS	— a grafikus kurzor *Y* magasságát tárolja.
GD	— a grafikus kurzor üzemmódja (állapota). Ha értéke 1, akkor kurzorszínű, ha 0, akkor alapszínű pontokkal jelenik meg.
FL$	— filenév tároló *LOAD*-nál és *SAVE*-nél.
XT	— kör vagy elipszis *X* tengelye.
YT	— kör vagy elipszis *Y* tengelye.
XE	— téglalap jobb alsó sarkának *X* koordinátája.
YE	— téglalap jobb alsó sarkának *Y* koordinátája.
FO	— elforgatási szög értéke.
P$	— kitöltés.
XV	— vonal végpontjának *X* koordinátája.
YV	— vonal végpontjának *Y* koordinátája.

A memória felépítése a program betöltése után

$1000—$1FFF Basic program egy része.
$2000—$3FFF Grafikus memória.
$4000—$5DE9 Basic főprogram.
$7000—$7CFF Karakterkészlet

Ötlet a továbbfejlesztéshez

Szöveges módban a kurzor nem villog, hanem egy helyben áll. Ha azt szeretnénk, hogy villogjon, akkor írjuk be, ill. át a következő sorokat:

255 GETK$ vagy 255 GETK$: SOUND1,900,2
256 IFK$= " " THEN 362
362 CHAR1,X,Y,"&":FORV=1TOx:NEXTV
363 CHAR1,X,Y," ":FORV=1TOx:NEXTV
364 GOTO255

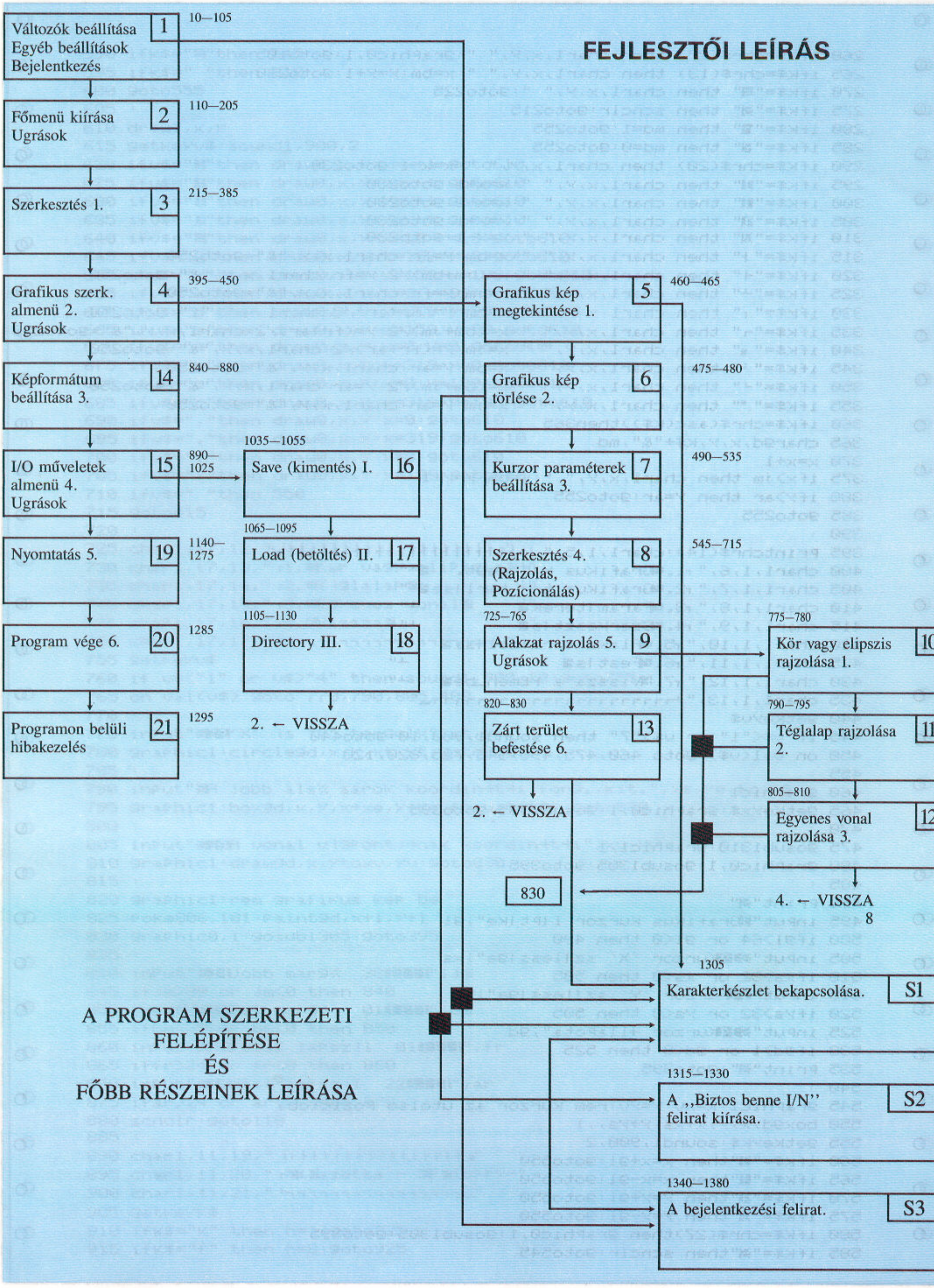
ahol x a kurzor villogásának sebessége. Szerintem x=50 a legjobb!

A kurzor alakját is átváltoztathatjuk, ha átlépünk a MONITOR-ba és átírjuk az $7530—$7537 címeket. Pl.:

7530 FF FF FF FF FF FF FF FF Hagyományos kurzort eredményez.
7530 FC FC FC FC FC FC FC FC Telt, de keskenyebb kurzort eredményez.

```
10 rem ************************************
15 rem * orszaGos commodore eGyesulet *
20 rem * alKalmazoi ProGram PalYazat. *
25 rem * ProGram : mini iratKeszito   *
30 rem * irta    : horvath Gabor      *
35 rem ************************************
40 :
45 traP1295:rem hibaKezeles
50 PoKe806,103:rem stoP KiKaPcsolva
55 h=8:aKtualis eGYseG a meGhajto
60 forf=1to8:KeYf,"":nextf:rem f-billentYuK "KiKaPcsolva"
65 :
70 bm=1:jm=38:fr=1:ar=22:rem KePformatum
75 Gl=1:xs=0:Ys=0:Gd=1:rem Grafikus Kurzor allaPota
80 :
85 GraPhic0,1
90 color1,1:color0,10,7:color4,10,7
95 Gosub1340
100 :
105 Gosub1305:rem uj KaraKtereK
110 Print"███";chr$(14);chr$(8):vol8
115 color1,7,0:color0,14,7:color4,7,0
120 char1,5,1,"█Horv4th G4bor:Mini IratKlszvtl█"
125 char1,8,4,"¦++++++++++++++++++++++++█"
130 for fm=1 to 14
135 char1,8,4+fm,"r                        +"
140 next fm
145 char1,8,19,"hrrrrrrrrrrrrrrrrrrrrrrr█"
150 char1,10,5,"F£men_"
155 char1,9,7,"1.█SzerKesztls█"
160 char1,9,9,"2.█Grafikus szerKesztls█"
165 char1,9,11,"3.█KlPform4tum█"
170 char1,9,13,"4.█I/O m@veleteK█"
175 char1,9,15,"5.█NYomtatv4nY Klszvtls█"
180 char1,9,17,"6.█KillPls Basic-be█"
185 sound1,850,10:sound1,900,10:sound1,950,10
190 GetKeYv$
195 :
200 if v$<"1" or v$>"6" then sound1,900,10:Goto190
205 on val(v$) Goto 215,395,840,890,1140,1285
210 :
215 GraPhic1:rem Grafikus KeP be
220 PoKe740,116:rem sajat KaraKtereK a Grafikus KePen
225 x=bm:Y=fr
230 ifx>jm then x=jm
235 ifx<bm then x=bm
240 ifY>ar then Y=ar
245 ifY<fr then Y=fr
250 char1,x,Y,"&",md
255 GetKeYK$:sound1,900,2
```

FEJLESZTŐI LEÍRÁS



A PROGRAM SZERKEZETI FELÉPÍTÉSE ÉS FŐBB RÉSZEINEK LEÍRÁSA

```
260 ifk$=chr$(27) then char1,x,y," ":graphic0,1:goto105
265 ifk$=chr$(13) then char1,x,y," ":x=bm:y=y+1:goto230
270 ifk$="■" then char1,x,y," ":goto225
275 ifk$="■" then scnclr:goto215
280 ifk$="■" then md=1:goto255
285 ifk$="■" then md=0:goto255
290 ifk$=chr$(20) then char1,x,y," ":x=x-1:goto230
295 ifk$="■" then char1,x,y," ":x=x+1:goto230
300 ifk$="■" then char1,x,y," ":x=x-1:goto230
305 ifk$="■" then char1,x,y," ":y=y+1:goto230
310 ifk$="■" then char1,x,y," ":y=y-1:goto230
315 ifk$="┣" then char1,x,y," ":x=bm:y=fr:char1,x,y,"&":goto250
320 ifk$="┫" then char1,x,y," ":x=(jm+bm)/2:y=fr:char1,x,y,"&":goto250
325 ifk$="┻" then char1,x,y," ":x=jm:y=fr:char1,x,y,"&":goto250
330 ifk$="┌" then char1,x,y," ":x=bm:y=(fr+ar)/2:char1,x,y,"&":goto250
335 ifk$="┐" then char1,x,y," ":x=(bm+jm)/2:y=(fr+ar)/2:char1,x,y,"&":goto250
340 ifk$="▪" then char1,x,y," ":x=jm:y=(fr+ar)/2:char1,x,y,"&":goto250
345 ifk$="┗" then char1,x,y," ":x=bm:y=ar:char1,x,y,"&":goto250
350 ifk$="┛" then char1,x,y," ":x=(bm+jm)/2:y=ar:char1,x,y,"&":goto250
355 ifk$="▀" then char1,x,y," ":x=jm:y=ar:char1,x,y,"&":goto250
360 ifk$=chr$(asc(k$))then365
365 chargd,x,y,k$+"&",md
370 x=x+1
375 ifx>jm then char1,x,y," ":x=bm:y=y+1
380 ify>ar then y=ar:goto255
385 goto255
390 :
395 printchr$(14):char1,1,5,"┃┃┃┃┃┃┃┃┃┃┃┃┃┃┃┃┃┃┃┃┃┃┃┃▪"
400 char1,1,6,"r1.■Grafikus kép megt.    ■┻"
405 char1,1,7,"r2.■Grafikus tőr tűrlése■┻"
410 char1,1,8,"r3.■Paraméterek■          ┻"
415 char1,1,9,"r4.■Szerkesztés■          ┻"
420 char1,1,10,"r5.■Alakzat rajzolása■   ┻"
425 char1,1,11,"r6.■Festés■              ┻"
430 char1,1,12,"r7.■Vissza a főmenübe■   ┻"
435 char1,1,13,"┗┳┳┳┳┳┳┳┳┳┳┳┳┳┳┳┳┳┳┳┳┳┳┳▬"
440 getkeyv$
445 if v$<"1" or v$>"7" then sound1,900,10:goto440
450 on val(v$) goto 460,475,490,545,725,820,120
455 :
460 graphic1
465 getkeyx$:graphic0,1:gosub1305:goto395
470 :
475 gosub1310:graphic1,1
480 graphic0,1:gosub1305:goto395
485 :
490 print"■"
495 input"■Grafikus kurzor lépték";gl
500 ifgl>64 or gl<0 then 490
505 input"■■Kurzor 'X' szélesség";xs
510 ifxs>32 or xs<0 then 505
515 input"■■Kurzor 'Y' szélesség";ys
520 ifys>32 or ys<0 then 505
525 input"■■Kurzor állapota";gd
530 ifgd>1 or gd<0 then 525
535 print"■":goto395
540 :
545 graphic1:x=xv:y=yv:rem kurzor az utolso pozicioba
550 boxgd,x,y,x+xs,y+ys,,1
555 getkeyk$:sound1,900,2
560 ifk$="■"then x=x+gl:goto550
565 ifk$="■"then x=x-gl:goto550
570 ifk$="■"then y=y+gl:goto550
575 ifk$="■"then y=y-gl:goto550
580 ifk$=chr$(27)then graphic0,1:gosub1305:goto395
585 ifk$="■"then scnclr:goto545
```

```
590 ifk$="█"then 545
595 ifk$=" "then 610
600 goto555
605 :
610 drawl,x,y
615 getkeyv$:sound1,900,2
620 ifv$="█"then draw0,x,y:x=x+1:goto610
625 ifv$="█"then draw0,x,y:x=x-1:goto610
630 ifv$="█"then draw0,x,y:y=y+1:goto610
635 ifv$="█"then draw0,x,y:y=y-1:goto610
640 ifv$="█"then draw0,x,y:x=10:y=10:goto610
645 ifv$="w"then draw0,x,y:x=159:y=0:goto610
650 ifv$="x"then draw0,x,y:x=159:y=199:goto610
655 ifv$="a"then draw0,x,y:x=0:y=99:goto610
660 ifv$="d"then draw0,x,y:x=319:y=99:goto610
665 ifv$="s"then draw0,x,y:x=159:y=99:goto610
670 ifv$="q"then draw0,x,y:x=0:y=0:goto610
675 ifv$="e"then draw0,x,y:x=319:y=0:goto610
680 ifv$="z"then draw0,x,y:x=0:y=199:goto610
685 ifv$="c"then draw0,x,y:x=319:y=199:goto610
690 ifv$=","then draw0,x,y:x=0:goto610
695 ifv$="."then draw0,x,y:x=319:goto610
700 ifv$=";"then draw0,x,y:y=0:goto610
705 ifv$="/"then draw0,x,y:y=199:goto610
710 ifv$=" "then 550
715 goto615
720 :
725 char1,17,12," ┌──────────────────┐"
730 char1,17,13," │1.█kör vagy ellipszis█│"
735 char1,17,14," │2.█Téglalap█          │"
740 char1,17,15," │3.█Egyenes vonal█     │"
745 char1,17,16," │4.█Vissza.█           │"
750 char1,17,17," └──────────────────┘"
755 getkeyv$
760 if v$<"1" or v$>"4" then sound1,900,10:goto755
765 on val(v$) goto 770,790,805,400
770 :
775 input"██X' és 'Y' tengely";xt,yt
780 graphic1:circle9d,x,y,xt,yt:goto830
785 :
790 input"██Jobb alsó sarok koordinátái,forg.,kit.";xe,ye,fo,p$
795 graphic1:box9d,x,y,x+xe,y+ye,fo,val(p$):goto830
800 :
805 input"██A vonal végpontjának koordinátái";xv,yv
810 graphic1:draw9d,x,ytoxv,yv:goto830
815 :
820 graphic1:rem grafikus kep be
825 poke806,101:paint9d,x+1,y+1
830 graphic0,1:gosub1305:goto395
835 :
840 input"██Jobb margó  38████";jm
845 ifjm>39 or jm<0 then 840
850 input"██Bal margó   01████";bm
855 ifbm>39 or bm<0 then 850
860 input"██Felső lapszél  01████";fr
865 iffr>24 or fr<0 then 860
870 input"██Alsó lapszél  22████";ar
875 ifar>24 or ar<0 then 870
880 scnclr:goto110
885 :
890 char1,11,19," ┌───────────────┐"
895 char1,11,20," │█Kazetta - █Floppy│"
900 char1,11,21," └───────────────┘"
905 getkeyk$
910 ifk$="k" then h=1:goto925
915 ifk$="f" then h=8:goto925
```

```
920 goto905
925 char1,6,5,"▄ ┼┼┼┼┼┼┼┼┼┼┼┼┼┼┼┼┼┼┼┼┼┼┼┼┼┼┼┼┼┼▄"
930 char1,6,6,"┌                             ┴"
935 char1,6,7,"┌                             ┴"
940 char1,6,8,"└┐┐┐┐┐┐┐┐┐┐┐┐┐┐┐┐┐┐┐┐┐┐┐┐┐┐┐┐┐▬"
945 color1,7,2:char1,7,6,"Save Load                    Vissza"
950 if h=1 then color1,7,4:goto960
955 if h=8 then color1,7,2:goto960
960 char1,17,6,"Directory"
965 o=9:rem nyil alap oszlopszama
970 ifo>32 or o<7 then o=7
975 char1,o,7,"▓←"
980 getkeyv$
985 ifv$="▓" then char1,o,7," ":o=o+1:goto970
990 ifv$="▓" then char1,o,7," ":o=o-1:goto970
995 ifv$=" " then color1,1:goto1005
1000 goto975
1005 ifo=8 then 1035
1010 ifo=13 then 1065
1015 ifo=18 then 1105
1020 ifo=28 then 110
1025 goto980
1030 :
1035 poke806,101:input"▓▓Filenev";f1$
1040 if len(f1$)>13 or len(f1$)<1 then 1035:else f1$=f1$+".gr"
1045 key1,"s"+chr$(34)+f1$+chr$(34)+","+str$(h)+",2000,4000"+chr$(13)
   +"x"+chr$(13)+"run"+chr$(13)
1050 print"▓Nyomja le az 'f1' gombot!"
1055 monitor
1060 :
1065 input"▓▓Filenev";f1$
1070 if len(f1$)>13 or len(f1$)<1 then 1065
1075 graphic1,1:graphic0:gosub1305:printchr$(14);"▓":char1,1,24,"▓Tolt is...▓"
1080 load f1$+".gr",h,1
1085 ifds>0 then 1090
1090 char1,1,1,""+ds$
1095 getkeyk$:graphic0,1:gosub1305:goto110
1100 :
1105 ifh=1 then 1125
1110 print"▓▓":directory
1115 ifds>0 then char1,1,1,""+ds$:getkeyk$:goto110
1120 getkeyk$:goto925
1125 print"▓▓Szalagrol nem lehet directoryt kirni!"
1130 getkeyk$:goto925
1135 :
1140 print"▓▓Tegye be a lemezt a meghajtoba amely"
1145 print"▓a karakterkeszletet tartalmazza!"
1150 print"▓▓▓▓Nyomjon le egy billentyut!"
1155 getkeyk$
1160 poke65298,196:poke65299,208:rem eredeti karakterek be
1165 graphic1:rem grafikus kep be
1170 open1,4:print chr$(8):x=326
1175 for i=1 to 46
1180 x=x-7:a$="":a1$="":b$="":b1$=""
1185 for y=0 to 199
1190 a=0:b=0
1195 locate x,y:if rdot(2)=1 then a=a+3
1200 locate x-1,y:if rdot(2)=1 thena=a+12
1205 locate x-2,y:if rdot(2)=1 thena=a+48
1210 locate x-3,y:if rdot(2)=1 thena=a+64:b=b+1
1215 locate x-4,y:if rdot(2)=1 thenb=b+6:if i=46 then 1230
1220 locate x-5,y:if rdot(2)=1 thenb=b+24
1225 locate x-6,y:if rdot(2)=1 thenb=b+96
1230 a=a+128:b=b+128
1235 p$=chr$(a):q$=chr$(b):if y>99 then1245
1240 a$=a$+p$+p$:b$=b$+q$+q$:goto1250
```

```
1245 a1$=a1$+p$+p$:b1$=b1$+q$+q$
1250 next y
1255 print#1,chr$(8)a$,a1$
1260 print#1,chr$(8)b$,b1$
1265 next i
1270 print chr$(15):close1
1275 load"karakterkeszlet",h,1
1280 :
1290 print"■":end
1295 resume next
1300 :
1305 poke65298,192:poke65299,112:return:rem uj karakterek
1310 :
1315 char1,16,10," ⌐┼┼┼┼┼┼┼┼┼┼┼┼┼┼┼┼┼*"
1320 char1,16,11," ⌐Biztos benne? I/N-"
1325 char1,16,12,"└┐┐┐┐┐┐┐┐┐┐┐┐┐┐┐┐┐▬"
1330 getkeyv$:ifv$="i" then return:else395
1335 :
1340 gosub1305:printchr$(14);chr$(8)
1345 char1,6,5,"ORSZ■GOS COMMODORE EGYES_LET"
1350 char1,6,7,"ALKALMAZ%I PROGRAM P■LY■ZATA"
1355 char1,12,10,"Mini Iratklszvt£"
1360 char1,4,12,"A programot vrta : HORV■TH G■BOR"
1365 char1,4,14,"COMMODORE PLUS/4 ls b£vvtett C16"
1370 char1,2,22,"COPYRIGHT (C)1993. DUNGEON SOFTWARE"
1375 forv=1to1500:nextv
1380 return

ready.
```

Ez a program egy ügyességi játék, amelyben egy vonalat kell úgy kormányoznunk, hogy ne ütközzünk neki semminek.

```
100 REM ******************************
110 REM *        RAJZOLO        C-64 *
120 REM *          , ,               *
130 REM *     KESZITETTE: GABRYEL    *
140 REM ******************************
150 :
160 FOR I=49152 TO 49205
170 READ A:POKE I,A:S=S+A
180 NEXT
190 IF S<>7941 THEN PRINT"HIBA"
200 DATA 120,165,001,072,041,251,133
210 DATA 001,169,000,133,251,169,208
220 DATA 133,252,169,000,133,253,169
230 DATA 048,133,254,160,000,162,000
240 DATA 177,251,145,253,200,208,249
250 DATA 232,230,252,230,254,224,016
260 DATA 208,240,104,133,001,169,028
270 DATA 141,024,208,088,096
280 SYS49152:POKE53272,28
290 FORI=0TO7:READA:POKE12288+I,A:NEXT
300 DATA 159,159,159,0,249,249,249,0
310 PRINT"■□":POKE53280,.:POKE53281,9
320 PRINTTAB(15)"■■■■■RAJZOLO■■
330 PRINTTAB(14)"■■K┐,■■ESZ┐,■■ITETTE :
340 PRINTTAB(15)"■GABRYEL
350 PRINTTAB(14)"■■■ SZABALY
360 PRINTTAB(14)"■■  JATEK
370 FORI=.TO39:POKE1024+I,.:POKE1064+I,
    .:POKE1984+I,.:POKE1944+I,.:NEXT
380 POKE53269,0:POKE54296,0
390 FORI=.TO24:POKE1024+40*I,
    .:POKE1025+40*I,.:POKE1062+40*I,.
400 POKE1063+40*I,.:NEXT
410 Q=19:B=128:W=22:GOSUB 470
420 J=PEEK(56320)
430 IF(JAND1)=0THENGOSUB470
440 IF(JAND2)=0THENGOSUB530
450 IF(JAND16)=0THEN580
460 GOTO 420
470 PRINT"■■■■■■■■■■■■■■■■■■■■■■■■"TAB(14);
480 PRINT"■ SZABALY "
490 PRINT"■■■■■■■■■■■■■■■■■■■■■■■■"
500 PRINTTAB(14)"  JATEK  "
510 A=1:RETURN
520 :
530 PRINT"■■■■■■■■■■■■■■■■■■■■■■■■";
540 PRINTTAB(14)" SZABALY "
550 PRINT"■■■■■■■■■■■■■■■■■■■■■■■■"
560 PRINTTAB(14)"■ JATEK  "
570 A=0:RETURN
580 :
590 IF A=0 THEN 660
600 PRINT"□■■"
610 PRINT"  A JATEKBAN EGY VONALAT KELL UGY
620 PRINT"■  IRANYITANI ,HOGY
    NE UTKOZZUNK NEKI
621 PRINT"■  SEMMINEK. IRANYITAS : 2.PORT
630 PRINT"■  JO SZORAKOZAST !
640 FORI=0TO1000:NEXT
650 IFPEEK(56320)=127 THEN 650
660 PRINT"□":A=0
670 :
680 FORI=0TO39:POKE1024+I,.
690 POKE 1984+I,.:NEXT
700 FORI=0TO24:POKE1024+40*I,.
710 POKE1063+40*I,.:NEXT
720 A=1563:Q=40
730 J=PEEK(56320):IF(JAND1)=0THENQ=-40
740 IF(JAND2)=0THENQ=40
750 IF(JAND4)=0THENQ=-1
760 IF(JAND8)=0THENQ=1
770 A=A+Q:P=P+1:IFPEEK(A)<>32THEN790
780 POKE54272+A,14:POKEA,0:GOTO730
790 :
800 PRINT"■■■■■■■■■■■■"TAB(11)
    "VEGE A JATEKNAK"
810 FORI=0TO2000:NEXT
820 PRINT"□■■■■■■■■■■■■■■■■■PONTOD :"P
830 FORI=0TO2000:NEXT
840 GOTO 310
```

Még többet ésszel!

XXI. rész

A sorozat eddigi részeiben nem egyszer sajnálattal vettem észre, hogy a cikkek szövegében a nyomdai előkészítés során kisebb-nagyobb sajtóhibák csúsznak, néha egész sorok maradnak ki. Mivel a kézirat leadása után a szöveggel a megjelent újságban találkozom legközelebb, a hibákat megelőzni nem tudom. Kiigazításukkal egy későbbi részben azért nem próbálkozom, mert az csak több hónappal a hiba megjelenése után lenne olvasható. Most kivételt teszek, talán nem feleslegesen, mert a XV. rész (93/9. 19. o.) három komoly hibát is tartalmaz. A 3. hasáb 3. bekezdésében „A LORD parancs ugyanúgy működik, mint azt a kazetta használatánál elmondtam." A 4. hasáb 4. bekezdésben a filenév: "$A?.?X"; a 6. bekezdésben a típusok DEL és SEQ.

A korábbi részekben többször is utaltam a soros állományok írásának és olvasásának szabályaira, anélkül, hogy elmondtam volna azokat. Azért tettem így, mert ezekkel többnyire mindenki eléggé tisztában van. Most terítékre kerül a téma, így legalább visszamenőleg minden tisztázható.

A lemezen négyféle soros állomány létezik: törölt (DEL), szekvenciális (SEQ), user (júzer, azaz felhasználói, USR) és program (PRG). Mindegyiküket lehet az OPEN utasítással írásra vagy olvasásra megnyitni, de ezenkívül a PRG típus a LORD és SAVE utasításokkal is kezelhető. A DEL típus, nevével ellentétben egy létező, bejegyzett file-t jelez, a valóban törölt file-nak nincs típusa. A DEL és USR típusok célja előttem, de tapasztalatom szerint mások előtt is ismeretlen, mivel a típus jelzésén kívül mindenben megegyeznek a SEQ típus jellemzőivel.

Van az adattárolásban három olyan alapfogalom, amelyeket talán már most bemutathatok. A **file**, magyarul állomány már érthetővé válhatott az eddigiek alapján. Az állományba összetartozó adatokat helyezünk el, egyetlen csomagba, egy közös név alá. Elsősorban adatállományok (tehát más főprogram által létrehozott és kezelt adatok) esetében értelmezhető a **rekord** fogalma. Ha az adatállományban személyi adatokat tárolunk, akkor az egy személyre vonatkozó adatok vannak egy rekordban. A rekordok összessége adja az állományt. Egy BASIC programállományban rekordnak nevezhetjük egy programsor kódolt adatait. A rekordok különböző hosszúak is lehetnek, a relatív file rekordjai viszont jellemzően azonos hosszúak. Ha a rekord hosszúsága előre nem ismert, akkor csak egy Rekord Vége Jel segítségével lehet őket elkülöníteni, amely megegyezés szerinti lehet, a gép e célra a Return (13-as kódú) karaktert használja.

Az adatrekordokban több különféle jelentésű, típusú, hosszú adat**mező** lehet, mint például a név, személyi szám, lakcím stb. A mezők a feldolgozás ésszerűsítése érdekében rekordonként azonos hosszúak szoktak lenni. Mivel kellően hozzáértő módon bármilyen adattárolási szerkezet megvalósítható, általános szabályok nincsenek, legfeljebb hagyományok és jobb vagy rosszabb megoldások.

A „soros állomány" megnevezést össze lehet keverni a SEQ, azaz szekvenciális típusú állománnyal, amelyet szintén, sorosnak fordíthatunk. Sorosnak hívjuk azokat az állománytípusokat, amelyeket csak sorosan, azaz a rekordokat egymás után lehet olvasni, írni, feldolgozni. Az ilyen állományból pl. a 100. rekordot csak úgy lehet beolvasni, ha már mind a 99 megelőző rekord olvasását elvégeztük. Nem szükséges persze ezeket mind felhasználni, értelmezni, csak az olvasással léptetjük azt a mutatót, amely a lemezegységben az állomány következő rekordjának helyét kijelöli. A 100. rekord kiolvasása után pl. a 70. rekordot csak úgy lehet ismét megtalálni, ha az állományt lezárjuk, újra megnyitjuk, és az első 69 rekordot újra kiolvassuk.

Látható, hogy ez a technika nem teszi lehetővé a rekordok közötti szabad ugrándozást. Ezért soros állományokban olyasmit szokás tárolni, amely adatokat egymás utáni sorrendben akarjuk feldolgozni. Van egy másik lehetőség is: a soros állomány összes rekordját nagyjából egyszerre beolvassuk, és a memóriában (tömbökben, változókban, közvetlenül) tároljuk. A memória már nem soros, hanem tetszőleges elérésű (ezt jelenti a RAM), tehát a betöltött állományban már felborítható a hozzáférési sorrend.

Ez a módszer különösen lényeges lehet akkor, ha a soros állományban akarunk javítani valamit. A DOS nem engedi meg, hogy az állományokat csak egy meghatározott szakaszukon írjuk felül, hacsak nem alkalmazunk közvetlen lemezműveleteket, amelyek már nem kapcsolódnak a soros állományok témájához. Különben is, többnyire nem pontosan ugyanolyan hosszú adatot (rekordot) akarunk visszaírni, mint amilyen hosszú a korábbi adat volt. És mivel a javítandót követő adatokat nem lehet „odébbtolni" a lemezen, így vagy hézag maradna a rekord után (ami azonban önálló rekordnak számítana), vagy ráírnánk a következő rekord elejére. Ez tehát nem megy. Marad a következő: az egész állományt betöltjük a tárba, a javításokat ott végezzük el, és a teljes állományt ismét egészben a lemezre mentjük, a régit törölve. Nehézkes módszer de a soros állományok gazdaságos helyfoglalása és az olvasás viszonylagos egyszerűsége néha megérhet egy kis többletmunkát. Ráadásul szalagos tárolókon nem is lehet más módszert választani. Sajnos a módszert alapvetően korlátozza a tár mérete: de a gép lassúsága is. Ugyanis ha a legegyszerűbb módon, egy karakteres tömb minden elemébe egy rekordot beolvasva akarjuk tárolni az állományt, komolyabb méretek esetén a hulladékgyűjtés a tömböt sok másodpercig fogja rendezgetni.

Minden file lényegében kétféle műveletre nyitható meg: írásra és olvasásra. A soros állományok egyszerre csak ezek egyikére. Valamikor még a XIII. részben elmondtam a nyitó utasítás szerkezetét, de lássuk még egyszer: **OPEN lf[,dn[,sa[,parm]]]**. Az **lf** a file száma, a **dn** az eszközt azonosítja. Az **sa** egy soros file nyitásakor 2 és 14 közötti értékeket kaphat. A 15-ös érték jelentését már megtárgyaltuk, a magasabb értékekkel pedig lehet kísérletezni, de „élesben" nem tanácsolom a használatukat. Ezzel a számmal egy ún. csatornát nyitunk a gép és a lemezegység között, meg egy puffert is lefoglaltatunk az átmenő adatok számára. Ha a számot a fenti tartományból választjuk, akkor nincs semmi hatása a file típusára, tartalmára; a számok ilyen szempontból egyenrangúak. Csak arra kell ügyelni, hogy sem az **lf**, sem az **sa** nem lehet két egyidőben nyitott file-nál azonos.

Soros állománynál a **parm**, amelyet idézőjelbe kell írni, három fontos adatot is tartalmazhat: a file nevét, típusát és a megnyitási módot. Ezeket egymástól vesszővel kell elválasztani. A filenév legalább egy, legfeljebb tizenhat karakter hosszú lehet, néhány tiltott karakter (pl. vessző, kettőspont, egyenlőségjel, csillag, kérdőjel) kivételével bármit tartalmazhat. Ha egy már létező állomány helyett egy másikat akarunk a lemezre írni, akkor ezeken felül a név elejére oda lehet írni a „@:" jelzést is,

ahogy azt az előző részben elmondtam. Ennek a jelzésnek olvasásra megnyitott file-nál egyébként nincs semmi hatása. Olvasásra megnyitáskor globális név is megadható, ekkor a feltételeknek megfelelő nevű állományok közül az első lesz megnyitva.

A file típusát adhatjuk meg a **parm** második elemeként. Bármit is írunk ide, csak az első karaktert figyeli a DOS, ez pedig a következő lehet: D, S, P, U. Valóban, ezek a típusok rövidítéseinek első betűi. Tulajdonképpen nincs semmilyen megkötés arra, hogy melyik típust mire használjuk, mindegyik ugyanúgy alkalmas adattárolásra. Ha egy létező állományt akarunk olvasni, és megnyitásakor más típust jeleztünk, mint a file igazi típusa, 64-es hibaüzenetet kapunk. Ha olvasáskor a típus helyett csillagot írunk, akkor a DOS csak a név alapján azonosítja a file-t, és bármilyen típusú is az, megnyitja. Sőt, ha nem írunk be típust, vagy olyat adunk meg, amely nem is létezhet, akkor ezeket ugyanígy kezeli. Írásra megnyitásnál mindig egyértelmű és szabályos adatokat kell megadnunk. Az érvénytelen típust a DOS ,,S''-sel helyettesíti. Végül meg kell jegyeznem, hogy két azonos nevű file még eltérő típussal sem hozható létre egy lemezen.

A harmadik elem a megnyitási mód. Ennek is csak az első betűje lényeges, soros állománynál négyféle lehetséges: W (write, írás), A (append, folytatás), R (read, olvasás), M (jelentését nem ismerem). Az írás és olvasás talán nem igényel magyarázatot. Olvasni csak már létező állományt lehet, írni pedig csak még nem létezőt. Ha a mód jelét elhagyjuk, akkor a megnyitás olvasásra történik.

Az ,,A'' mód írást jelent, de nem új file létrehozását, hanem már meglevő file folytatását. Az ilyen megnyitást követően a kiírt adatok a file végéhez illesztve, folytatólagosan kerülnek a lemezre.

Az ,,M'' módról nem minden könyv tesz említést. Olvasást jelent, de olyan állományokból (is), amelyek íráskor nem lettek szabályosan lezárva. Az ilyen filenevek mellett a tartalomjegyzékben egy csillag látszik. Ez a file csak kézi munkával zárható már le (a korábban elmondottakat hasznosítva), és olvasásra, folytatásra (,,R'' és ,,A'') nem nyitható meg (60-as hibaüzenet). Írásra megnyitva a lezáratlan file megsemmisül, eltűnik, de az általa felhasznált terület nem szabadul fel. Szabálytalan lezárás nem csak kezelői hiba következtében fordulhat elő, hanem például egy áramszünet hatására is. A lemezre kiírt adatok ezután még egy ,,M'' módban végzett olvasás során kinyerhetők, más néven újra kiírva megmenthetők. A lezáratlan állományokat a validálás törli.

Hódi Gyula

Tippek—Trükkök

A következő program kis hangeffekteket állít elő a COMMODORE 64-es gépeken.

```
890 @
10 REM *** HANGEFFEKTEK - GABRIEL ***
100 PRINT"🗋"
110 POKE53280,.:POKE53281,9:POKE646,0
120 FORI=0TO24:POKE54272+I,.:NEXT
130 GOSUB910:Q=54272
140 POKE54278,245:POKE54275,8:POKEQ,0
150 POKE54296,15:POKEQ+1,.
160 GETA$:IFA$=""THEN160
170 A=ASC(A$)-65
180 ONA+1GOSUB200,250,310,390,470,
    560,630,700,740,810,900
190 GOTO 140
200 :
210 POKE 54276,65
220 FORI=20TO60STEP5:POKE54273,I
230 FORU=0TO10:NEXT:NEXT
240 POKE54273,0:RETURN
250 :
260 POKE54276,129
270 FORI=0TO10:POKE54273,10
280 FORU=0TO10:NEXT:POKE54273,0
290 FORU=0TO80:NEXT:NEXT
300 RETURN
310 :
320 POKE54276,65
330 FORI=60TO20STEP-2:POKE54273,I
340 FORU=0TO75:NEXT:NEXT
350 POKE54276,129:POKE54273,10
360 FORI=15TO0STEP-1:POKE54296,I
370 FORU=0TO30:NEXT:NEXT
380 RETURN
390 :
400 POKE54276,65
410 FORI=9TO255:POKE54272,I:NEXT
420 POKE54273,1
430 FORI=0TO255:POKE54272,I:NEXT
440 FORI=15TO0STEP-1:POKE54296,I
450 FORU=0TO100:NEXT:NEXT
460 RETURN
470 :
480 POKE54276,65
490 POKE54273,1:POKE54272,255
500 FORI=0TO15:POKE54296,I
510 FORU=0TO100:NEXT:NEXT
520 FORI=255TO0STEP-1:POKE54272,I:NEXT
530 POKE54273,0
540 FORI=255TO0STEP-1:POKE54272,I:NEXT
550 RETURN
560 :
570 POKE54276,65
580 FORI=0TO15:POKE54296,I
590 FORU=20TO40:POKE54273,U:NEXT:NEXT
600 FORI=0TO20:POKE702,I
610 FORU=20TO40:POKE54273,U:NEXT:NEXT
620 RETURN
630 :
640 POKE54276,65
650 FORI=20TO40:POKE54273,I
660 FORU=0TO1:NEXT:NEXT
670 FORI=40TO20 STEP-1:POKE54273,I
680 FORU=0TO1  :NEXT:NEXT
690 RETURN
700 :
710 POKE54276,65
720 FORI=60TO10STEP-1:POKE54273,I:NEXT
730 RETURN
740 :
750 POKE 54276,129
760 FORI=0TO30:POKE54273,20
770 FORU=0TO50:NEXT
780 POKE54273,0
790 FORU=0TO50:NEXT:NEXT
800 RETURN
810 :
820 POKE 54276,65
830 FORO=0TO4
840 FORI=0TO10:POKE54273,20
850 FORU=0TO20:NEXT
860 POKE54273,0
870 FORU=0TO20:NEXT:NEXT
880 FORI=0TO1000:NEXT
```

Felhasználói programok

Lemezes gyorstöltők

A VC1541-es soros átvitelű lemezegységek lassúsága megjelenésük óta boszszantotta a COMMODORE tulajdonosokat. Igaz ugyan, hogy a magnónál lényegesen gyorsabb a floppy-drive, de ezt is éppúgy lehet turbózni, mint a magnót. Szerencsére C64-en sok gyorstöltőt készítettek a PLUS/4 megjelenések idejére, így elég gyorsan megkezdődhetett a C64-es tapasztalatok alapján a PLUS/4-es gyorstöltők írása.

Úgy hiszem, rögtön az elején definiálnom kell néhány alapvető fogalmat. Ezek: szektor-lépésköz, szektor-töltő, track-töltő.

A lemezegység egyik fontos változója a szektor-lépésköz. Azt adja meg, hogy hány szektorhely legyen a lemezre kerülő program egyes blokkjai között. Értékét a lemezegység belső változója határozza meg, mely a $69-es címen található. Alapesetben értéke 10 a 1541-es modelleken, és 5 a 1551-eseken. Ha például programunk a 17. track 0. szektorán kezdődik, akkor 10-es lépésköz esetén a következő blokk a 17. track 10. szektora lesz, és így tovább.

Szektor-töltőnek nevezzük azokat a gyorstöltőket, amelyek a tárba egyszerre csak a töltendő program egy bokkját viszik be, majd megkeresik a következőt, azt is beviszik, és így tovább a program végéig. Előnyük a kis helyfoglalás, gyakran elférnek a rendszerváltozó területen, hátrányuk a korlátozott működési sebesség. Egy szektor beolvasása után ugyanis átküldik a gépnek. Ha ezalatt több idő telik el, mint amennyi alatt a program következő blokkját tartalmazó szektor az olvasófej elé jut, akkor a gyorstöltőnek meg kell várnia, míg a lemez egyszer teljesen körbefordul, és így a szektor újból elérhető lesz. A PLUS/4-es gyorstöltők nagy része ebbe a kategóriába tartozik.

A track-töltők ezzel szemben egyszerre egy teljes sávot olvasnak be a gép tárába, méghozzá a nekik legjobban megfelelő szektor-lépéssel haladva, ami esetleg nem egyezik meg a felírási lépésközzel. Az igen időigényes úgynevezett GCR kódolást a gép végzi el saját memóriájában, ugyanis a gép processzora a magasabb órajel következtében a floppyénál gyorsabban fut, így a kódolás is kevesebb ideig tart. Ez a kódolás a meghajtó belső felépítése miatt szükséges, pontosabban nem térek ki rá. Tehát a gép megkapja a sáv anyagát, kikódolja, majd a programot az így a memóriába kerülő szektorokból tölti be. Ez a töltés tehát nem más, mint a memóriába töltött szektorok megfelelő sorbarendezése. A bonyolult működés következtében több helyet foglalnak a memóriában, de sokkal gyorsabban működnek, mint a szektoros töltők.

Az első gyorstöltő-program a HYPRALOAD+4 volt, amely 1986-ban tűnt fel. Betöltés után RUN-nal indul, és a képernyőn ki is írja nevét. Ha ezután a lemezegységről töltünk, a képet lekapcsolja (mint a gyorstöltők többsége), átküld egy rövid programot a lemezegységbe, betölti a programot, a drive-ot reseteli (ezért a VC1541/C sorozatú egységek töltés után a bekapcsoláskor jelentkező kattogást hallatják), majd beadja a BASIC képernyőt. A betöltendő program nem lehet hosszabb 234 blokknál, ugyanis a memória csak $F780-ig szabad, után a gyorstöltő helyezkedik el. A tesztként használt 207 blokk terjedelmű programot 10-es szektor-lépésköznél 25,2 másodperc; 9-esnél 26,9 másodperc; 8-asnál 28,8 másodperc alatt töltötte be. Nyolcasnál alacsonyabb lépésköz esetén a töltési sebesség nagyon visszaesik, ami a szektoros töltési mód következménye.

Talán az FLOAD sorozatba tartozó programok a legelterjedtebbek. Legutolsó verziószám a 3.2 volt, illetve ennek egy alfaja, a 3.22, amelyben a változás annyi, hogy a tölteni kívánt programot menüből lehet kiválasztani, mint a MEGALOAD V2-ben, melyről később szólok. A gyorstöltőalgoritmus Nagy Tamás munkája. Szektor-gyorstöltő ez is, a BASIC memóriából nem foglal el semmit, így $FD00-ig tölthetjük programjainkat. Indítás után a funkciógombokat átírja. A directoryban megjelenő program az F2-[RETURN] lenyomásával tölthető, F6-tal futtatható, F3-mal kérhetünk directoryt. A kísérleti programot csak 10-es szektor-lépésközzel tudta gyorsan betölteni, ami a rutin lassúságából ered, aminek visszafejtését egyszerűsége miatt kezdők számára is ajánlhatom. A töltési idő 23,7 másodperc volt. Töltés közben a képet lekapcsolja és csíkozza. Ezt a rutint használja töltésre elég sok lemezes játék (esetenként módosítva, amely általában a csíkozás kikapcsolásából áll), többek között a MUFFBUSTERS CREW programjai, például a HOLLYWOOD POKER PRO is (aki ezt nem hiszi, hasonlítsa össze a POKER PRO 4 blokkos file-jának $10CC—10FD-ig terjedő szakaszát az FLOAD 3.2 $119E—11C5 darabjával, illetve a POKER PRO $121D—1246 darabját az FLOAD $12A0—12C9 tartományával). Furcsa módon a csoport tagjai ezért sohasem mondtak köszönetet a gyorstöltő írójának.

Különlegesnek számít CNS gyorstöltője: bekapcsolt képpel tölt, és a töltési címet folyamatosan mutatja a képernyő jobb felső sarkában. Töltőalgoritmusát csak ínyenceknek ajánlom: ,,illegális" utasításokat használ, így a gépi TEDMON többször csak ??? jelet ír ki visszafejtéskor. A rutin egyedülálló: bár több bekapcsolt képpel működő gyorstöltőt láttam, mindegyik ezt a rutint használja (így például a ,,C" nevű program), valamint PIGMY több szöveges játéka (például a SCAPEGHOST). Tízes lépésköznél 25,1; kilencesnél 25,2; nyolcasnál 23,3 másodperc alatt töltötte be a tesztprogramot. Ennél kisebb lépésköz esetén rendkívül lelassul, mert ez is szektoros töltő.

PIGMY, a főként C64 átiratairól ismert programozó a gyorstöltők sorában is nagyot alkotott: megírta a MEGALOAD sorozatot. A MEGALOAD V1-hez sosem volt szerencsém. A V2-t, melynek hossza 8 blokk,

LOAD"M+",8,1

utasítással kell tölteni. Automatikusan indul, betölti a directoryt, majd egy színes sávval választhatunk a programok közül. Ha új lemezet teszünk a drive-ba, a lemeznevet vagy a ... BLOCKS FREE szöveget tartalmazó soron RETURN-t ütve betölti annak tartalomjegyzékét. Csak egyszeri betöltésre használható. Betöltés után a képernyő tetején a RUN parancs jelenik meg, a kurzor ezen villog, így a RETURN megnyomásával a program azonnal indítható. A példaprogramot tízes lépésközzel 23,7; kilencessel 23,3; nyolcassal 22,4 másodperc alatt töltötte be. Ez is szektoros töltő, a teljes memóriába képes tölteni, nyolcasnál alacsonyabb lépésköz esetén lelassul.

A V3.0 verzió a memóriából az $FA00-ig terjedő területet hagyja szabadon, és a V2-vel szemben többször is használható. Szektoros elven működik, töltési eredményei: 24,7 másodperc tízes

lépésközzel, 24,3 kilencesnél és 23,3 nyolcasnál. RUN/STOP+RESET után BASIC-be kilépve Fl-gyel ,,élesíthető", csakúgy, mint a 4.0 verzió.

A V4.0 már trackes töltő, C64-ről ered, ha jól tudom. A memóriát $F200-tól használja, de még így is szinte bármit betölthetünk vele (227—228 blokkos-programokat is). A szektor-lépésköz nem befolyásolja, a tesztprogramot lépésköztől függetlenül 14,4 és 16,6 másodperc alatt töltötte be (a különbség oka, hogy a program a lemez más és más részén helyezkedett el), ami körülbelül kétharmada az átlag szektoros töltés idejének. Töltés közben a lemezegység fejléptetését felgyorsítja.

A lemezegységen kisebb hardver beavatkozást elvégezve annak sebessége lenyűgöző mértékben emelkedik. Az átalakítás lényege, hogy a floppyt párhuzamos kábellel kapcsoljuk a géphez. A párhuzamosított gépekre PIGMY megírta a PORT-TURBO nevű gyorstöltőt, ami a MEGALOAD V4.0-t is messze felülmúlja. Ez is trackes gyorstöltő, és a tesztelésre használt 207 blokk hosszú programot 9,3 másodperc (!) alatt töltötte be. A memóriát $DD00-ig használhatjuk szabadon, de egyszer be lehet tölteni olyan programot is, mely ebbe éppen nem fér bele (ilyen volt az említett 207 blokkos program is), de ezután a turbót többet nem használhatjuk.

Hogy a sebességnövekedést illusztráljam, leközlöm a párhuzamos floppyn futó szektoros gyorstöltő adatait is: a PORTLOAD nevű program (ami az FLOAD párhuzamosított változata) bekapcsolt képpel képes hetes lépésközű program olvasására, 19,6 másodperc alatt töltve 207 blokkot.

Ennyit szerettem volna írni a gyorstöltőkről, remélem, mindenki megtalálja a neki legjobban megfelelőt a cikkben közölt adatok alapján. A következő cikkben a két legelterjedtebb file-másoló leírása következik: a C64-en is ismert DUPLICATOR-é, és a szintén C64-es eredetű FLASHCOPY-é. Viszontlátásra!

Kovács István Attila

Színes listázó

```
1 REM ********************************
2 REM *                              *
3 REM *        COLOR LIST            *
4 REM *                              *
5 REM ********************************
6 :
7 :
8 PRINT CHR$(147)"DATA BEOLVASAS ES ELLENORZES ...";:J=02049:VE=02845:P=J
9 FOR B=0 TO 7:READ A$
10 L=ASC(MID$(A$,2,1))
11 H=ASC(MID$(A$,1,1))
12 L=L-48:IF L>9 THEN L=L-7
13 H=H-48:IF H>9 THEN H=H-7
14 PRINT"███████████"P;:P=P+1
15 IF H>15 OR L>15 THEN 17
16 A=H*16+L:POKE J+B,A:T=T+A:NEXT B:READ A:IF A=T THEN 18
17 PRINT:PRINT"DATA HIBA ...  SOR:"PEEK(64)*256+PEEK(63):END
18 T=0:J=J+8:IF J<VE THEN 9
19 PRINT"██KESZ":END
20 DATA 17,08,C2,07,9E,20,32,30,0520
21 DATA 37,33,2C,43,4F,4C,4F,52,0533
22 DATA 20,4C,49,53,54,00,00,00,0348
23 DATA 20,F1,B7,A9,50,A2,08,85,1008
24 DATA F7,86,F8,A0,00,A9,33,A2,1171
25 DATA CD,85,F9,86,FA,A5,FA,C9,1587
26 DATA D0,D0,09,A5,F9,C9,00,D0,1248
27 DATA 03,4C,33,CD,B1,F7,91,F9,1153
28 DATA E6,F7,D0,02,E6,F8,E6,F9,1644
29 DATA D0,02,E6,FA,18,90,DE,A9,1249
30 DATA F7,A0,CD,8D,06,03,8C,07,0909
31 DATA 03,A9,AA,A0,CD,8D,08,03,0859
32 DATA 8C,09,03,A9,EE,A0,CD,8D,1065
33 DATA 02,03,8C,03,03,A9,DB,A0,0699
34 DATA CD,8D,00,03,8C,01,03,A9,0662
35 DATA 00,8D,20,D0,8D,21,D0,A9,0932
36 DATA 6F,A0,CD,4C,1E,AB,0C,0F,0780
37 DATA 07,01,00,0D,99,43,4F,4C,0396
38 DATA 4F,52,2D,4C,49,53,54,45,0591
39 DATA 52,20,56,31,0D,11,56,4F,0444
40 DATA 4E,20,4E,49,4B,4F,4C,41,0556
41 DATA 55,53,20,48,45,55,53,4C,0585
42 DATA 45,52,0D,28,43,29,20,4E,0422
43 DATA 48,2D,30,38,31,32,39,30,0425
44 DATA 2D,41,52,52,0D,00,20,73,0434
45 DATA 00,C9,9B,D0,0A,AE,6A,CD,1059
46 DATA 8E,86,02,A2,00,86,B6,C9,0957
47 DATA 21,F0,06,20,79,00,4C,E7,0739
48 DATA A7,4C,4B,CE,C9,0D,D0,0C,0958
```

A színes listázóval a BASIC-programokat áttekinthetőbbé tehetjük. Egy szempillantás alatt felismerhetőkké tehetjük pl. a REM-sorokat és az utasításokat. Aztán azért, hogy a programozásban ne feledkezzünk meg az időről se, a szoftverből működő óra állandóan mutatja a dátumot és a pontos időt.

Ez az ugyancsak szokatlan, de hasznos segédprogram a LIST BASIC-utsítást további tulajdonsággal bővíti. Az utasításszavak és a REM-ek mögé írott megjegyzések, kommentárok színesen íródnak ki a képernyőre. Emellett még egy kis formatáló eszközt is tartalmaz.

Ha hosszú programot írunk, ezt értelemszerűen több részre osztjuk fel, mondjuk pl. beadó-, osztályozó-, nyomtatórutinra stb. A programciklus is különféle szekciókból áll. Az első szekcióban a változókat definiáljuk, aztán jönnek az adatbeolvasások, amiket aztán a harmadik rész értékel ki, majd a negyedik részben az eredmények íródnak ki.

Akármennyire is fáradozunk a strukturált programozáson, amikor beadjuk a LIST-et, hogy az egyes sorokat kijavítsuk az egész program szinte elzúg előttünk a képernyőn, és az áttekinthetőség máris odavan. Egyes programozók emiatt REM-sorokat iktatnak közbe azért, hogy a program különböző szekcióit egymástól elválasszák, megkülönböztessék. Azonban az is nehéz dolog, hogy az egyes REM-sorokat az előttünk ,,elzúgó" hosszabb listában felismerjük, elolvassuk, és egyet se nézzünk el, ne hagyjunk ki.

A ,,színes listázó" rendkívüli módon megkönnyíti a hibakeresést, mert a listában a BASIC-tokenek (vagyis az utasításszavak, mint pl. a PRINT, GOTO és így tovább) színesen jelennek meg a képernyőn. Ugyanígy lesznek láthatók a REM mögötti magyarázatok, megjegyzések is. A színeket tetszés szerint lehet megválasztani. További lehetőség még, hogy az egyes utasításszavak mögé be lehet szúrni egy üres szóközt. Míg a színek ter-

mészetszerűen csak a képernyőn lesznek láthatók, addig a szóközök a kinyomtatott listán is megmaradnak.

A tisztán gépi kódban írott programot töltsük be úgy, mint a normál BASIC-programot a

LOAD "COLOR-LIST",8

paranccsal és indítsuk el RUN-nal. A program beíródik a memóriába, módosítja a LIST-utasítást, és egy cím-képernyővel bejelentkezik. Most be lehet tölteni a javítandó programot, vagy el lehet kezdeni a program beírását. A LIST-utasítás kiadásakor a lista a szokásos módon lesz látható a képernyőn, de színesen a következőképpen:

A lista maga (tehát pl. a sorszámok) középszürke színű lesz. Az utasításszavak világosszürke színűek, a REM-sorok pedig fehérek. A direkt módban beadott parancsok, valamint a gép üzenetei, pl. a READY, a hibajelzések és a felhasználó által beadott adatok sárgák lesznek.

Ha ezek a színek nekünk éppen nem tetszenek, egyszerűen változtathatók. Írjunk direkt módban a képernyőre egy felkiáltójelet (!), majd adjunk RETURN-t. Erre megjelenik egy menü, amiben a négy szín és az elválasztó szóköz kapcsolója lesz látható. Használjuk a számbillentyűket, amelyek az egyes színeket változtatják. Az ⟨1⟩ billentyű a lista színét kapcsolja át, a ⟨2⟩ az utasításszavak színét illeszti, a ⟨3⟩ a REM-sorok színét manipulálja, a ⟨4⟩ pedig a direkt módban beadott utasítások és az üzenetek színét kapcsolja át. Minden számbillentyű lenyomás eggyel továbblépteti az adott funkció színét 0—15 között. Az ⟨5⟩ billentyűvel a tokenek utáni szóköz létrehozását lehet be- és kikapcsolni. A funkciót a RETURN billentyű lenyomásával elhagyhatjuk és visszatérhetünk a direkt módba.

Még egy kiegészítés: Mivel nemcsak az utasításszavak, mint pl. a PRINT, GOSUB stb. tárolódnak tokenizálva a memóriában, hanem a matematikai és logikai jelek is, így azok is a tokenekkel azonos módon emelődnek ki a képernyőn. A színeket teljesen az ízlésünknek megfelelően választhatjuk ki. Mindenesetre ajánlatos, hogy a szöveg, ill. a képernyő betűszínét sötétebbre válasszuk, mint a tokenekét és a REM mögötti kommentárokét, hogy azok feltűnően „világítsanak" a listában.

Végül röviden elmagyarázzuk, hogy milyen belső folyamatok játszódnak le a színes listázó működése közben. A program a jelen 1. verzióban a memóriában az 52531—53247 ($CD33—$CFFF) címeken helyezkedik el, tehát nem foglal el egyetlen BASIC-memóriahelyet sem, sőt bőven hagy helyet további segédprogramok számára a 49152 címtől kezdődően. RESET-tel ki lehet szállni belőle, de mindaddig, amíg a memóriában van, SYS 52531 utasítással újra aktiválható. Nem várható, hogy valamely más programmal együtt alkalmazva probléma, ütközés merülne fel. Ha mégis, akkor a színes listázó lekapcsolható a következő két utasítással, amiket direkt módban így kettősponttal elválasztva kell ebben a sorrendben beadni:

SYS 65418:SYS 58451

A „színes listázó" működése a legfontosabb rendszervektorok elhajlításán alapszik. A LIST-vektor a program egy rutinjára mutat, ami a tokeneket és a REM-sorokat felismeri és ennek megfelelően színezi. A kimenet-vektor felismeri a sorvégeket és ha szükséges, lekapcsolja a REM-színezést. Az utasításvektor elhajlítása révén a LIST-parancs beadásakor bekapcsolódik az első jelzőszín, majd felismeri az új felkiáltójel-parancsot. A melegstart-vektor új rutinra mutat, ami az utasítás végrehajtása után beállítja a direkt-mód színét. A direkt-mód vektora el van hajlítva, és a direkt-mód színének a beállítása mellett arról is gondoskodik, hogy a kimenet vektora a ⟨RUN/STOP-RESTORE⟩-ral való kiszállás után az új értékre legyen elhajlítva. Ezáltal a „színes listázó" immunná, érzéketlenné válik ezen billentyűkombinációval szemben.

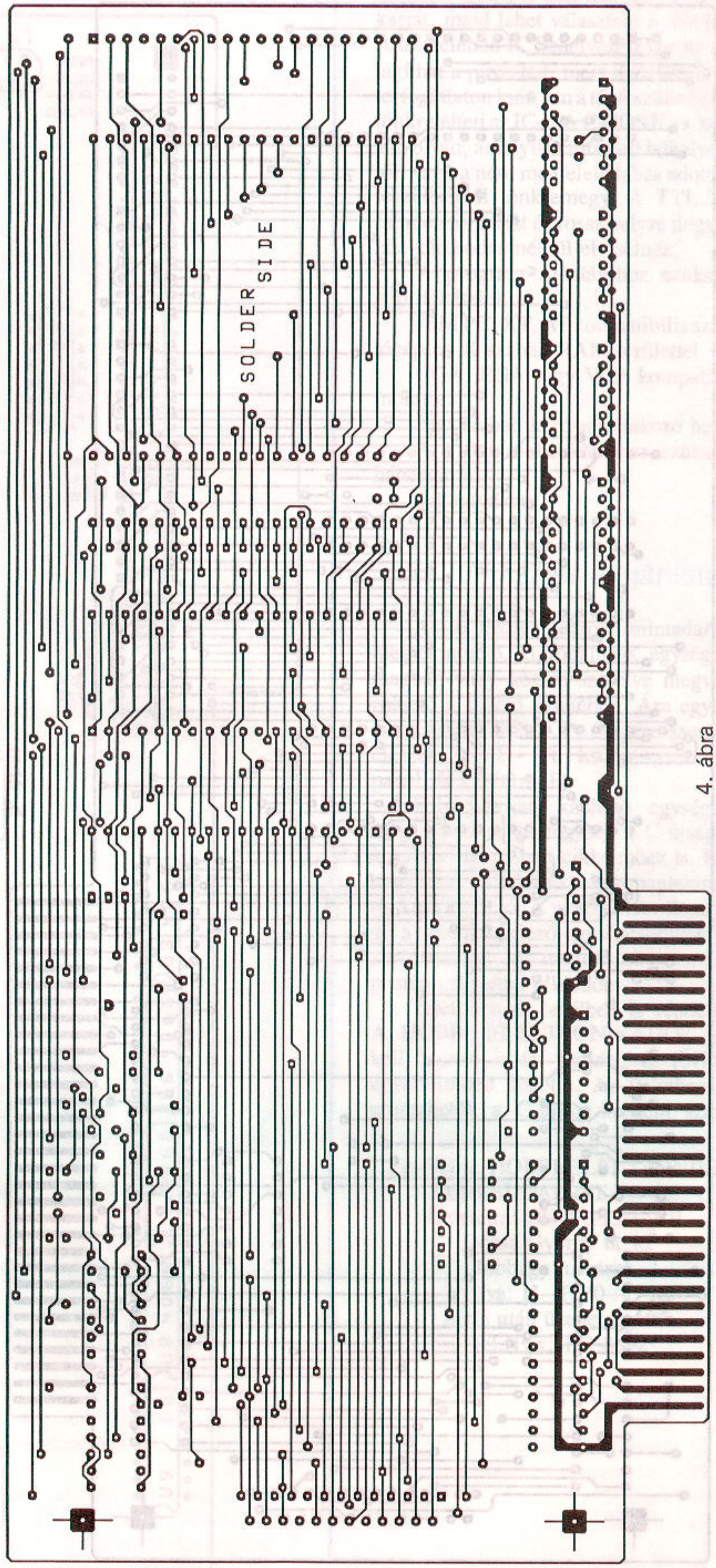
```
49 DATA A9,00,85,B6,AD,6A,CD,8D,1109
50 DATA 86,02,A9,0D,4C,CA,F1,AD,1010
51 DATA 6C,CD,8D,86,02,A9,C8,A0,1119
52 DATA CD,8D,26,03,8C,27,03,4C,0645
53 DATA 8B,E3,AD,6C,CD,8D,86,02,1129
54 DATA 4C,83,A4,A6,B6,D0,4B,AA,1172
55 DATA 30,09,AE,6A,CD,8E,86,02,0820
56 DATA 4C,F3,A6,C9,FF,F0,F3,24,1460
57 DATA 0F,30,EF,85,B5,38,E9,7F,1032
58 DATA AA,84,49,AC,6B,CD,8C,86,1133
59 DATA 02,A0,FF,CA,F0,08,C8,B9,1252
60 DATA 9E,A0,10,FA,30,F5,C8,B9,1262
61 DATA 9E,A0,48,29,7F,20,47,AB,0832
62 DATA 68,10,F3,A5,B5,C9,8F,D0,1261
63 DATA 02,E6,B6,AD,6E,CD,A4,49,1139
64 DATA D0,B8,AE,6D,CD,10,B6,A9,1247
65 DATA 2E,A0,CF,20,1E,AB,A9,0D,0828
66 DATA 85,D3,AE,6A,CD,20,05,CF,1073
67 DATA A9,0D,85,D3,AE,6B,CD,20,1044
68 DATA 05,CF,A9,0D,85,D3,AE,6D,1021
69 DATA CD,20,05,CF,A9,0D,85,D3,0975
70 DATA AE,6C,CD,20,05,CF,A9,0E,0914
71 DATA 85,D3,AD,6E,CD,F0,06,A9,1247
72 DATA 4E,A2,20,D0,04,A9,55,A2,0900
73 DATA 53,20,D2,FF,8A,20,D2,FF,1215
74 DATA A9,94,A0,CF,20,1E,AB,20,0949
75 DATA E4,FF,C9,0D,D0,10,A9,0D,1103
76 DATA A2,08,20,D2,FF,CA,D0,FA,1327
77 DATA 20,73,00,4C,AE,A7,C9,35,0818
78 DATA D0,0A,AD,6E,CD,49,20,8D,0952
79 DATA 6E,CD,10,8B,C9,31,D0,0E,0942
80 DATA EE,6A,CD,AD,6A,CD,29,0F,1089
81 DATA 8D,6A,CD,4C,4B,CE,C9,32,1060
82 DATA D0,0D,EE,6B,CD,AD,6B,CD,1256
83 DATA 29,0F,8D,6B,CD,10,EC,C9,0962
84 DATA 34,D0,0D,EE,6C,CD,AD,6C,1105
85 DATA CD,29,0F,8D,6C,CD,10,DB,0950
86 DATA C9,33,D0,A3,EE,6D,CD,AD,1348
87 DATA 6D,CD,29,0F,8D,6D,CD,10,0841
88 DATA CA,A0,FF,E8,CA,F0,08,C8,1499
89 DATA B9,9A,CF,30,F7,10,F8,C8,1305
90 DATA B9,9A,CF,48,29,7F,20,D2,1028
91 DATA FF,68,10,F3,A9,20,20,D2,1061
92 DATA FF,A6,D3,E0,25,90,F7,4C,1360
93 DATA D7,AA,0D,20,20,20,20,20,0558
94 DATA 20,20,54,45,58,54,3A,0D,0460
95 DATA 20,20,20,20,20,20,54,4F,0355
96 DATA 4B,45,4E,3A,0D,20,20,20,0389
97 DATA 20,20,20,20,20,52,45,4D,0388
98 DATA 3A,0D,20,20,20,20,20,52,0313
99 DATA 45,41,44,59,2E,3A,0D,4C,0484
100 DATA 45,45,52,5A,45,49,43,48,0591
101 DATA 45,4E,3A,20,41,0D,11,54,0416
102 DATA 41,53,54,45,4E,3A,20,3C,0529
103 DATA 31,3E,20,42,49,53,20,3C,0457
104 DATA 35,3E,20,55,4E,44,20,3C,0470
105 DATA 52,45,54,55,52,4E,3E,91,0687
106 DATA 91,91,91,91,91,00,53,43,0875
107 DATA 48,57,41,52,DA,57,45,49,0753
108 DATA 53,D3,52,4F,D4,54,55,45,0905
109 DATA 52,4B,49,D3,4C,49,4C,C1,0859
110 DATA 47,52,55,45,CE,42,4C,41,0720
111 DATA D5,47,45,4C,C2,4F,52,41,0849
112 DATA 4E,47,C5,42,52,41,55,CE,0850
113 DATA 48,45,4C,4C,52,4F,D4,44,0734
114 DATA 55,4E,4B,45,4C,47,52,41,0601
115 DATA D5,4D,49,54,54,45,4C,47,0747
116 DATA 52,41,D5,48,45,4C,4C,47,0724
117 DATA 52,55,45,CE,48,45,4C,4C,0735
118 DATA 42,4C,41,D5,48,45,4C,4C,0713
119 DATA 47,52,41,D5,35,0484

READY.
```

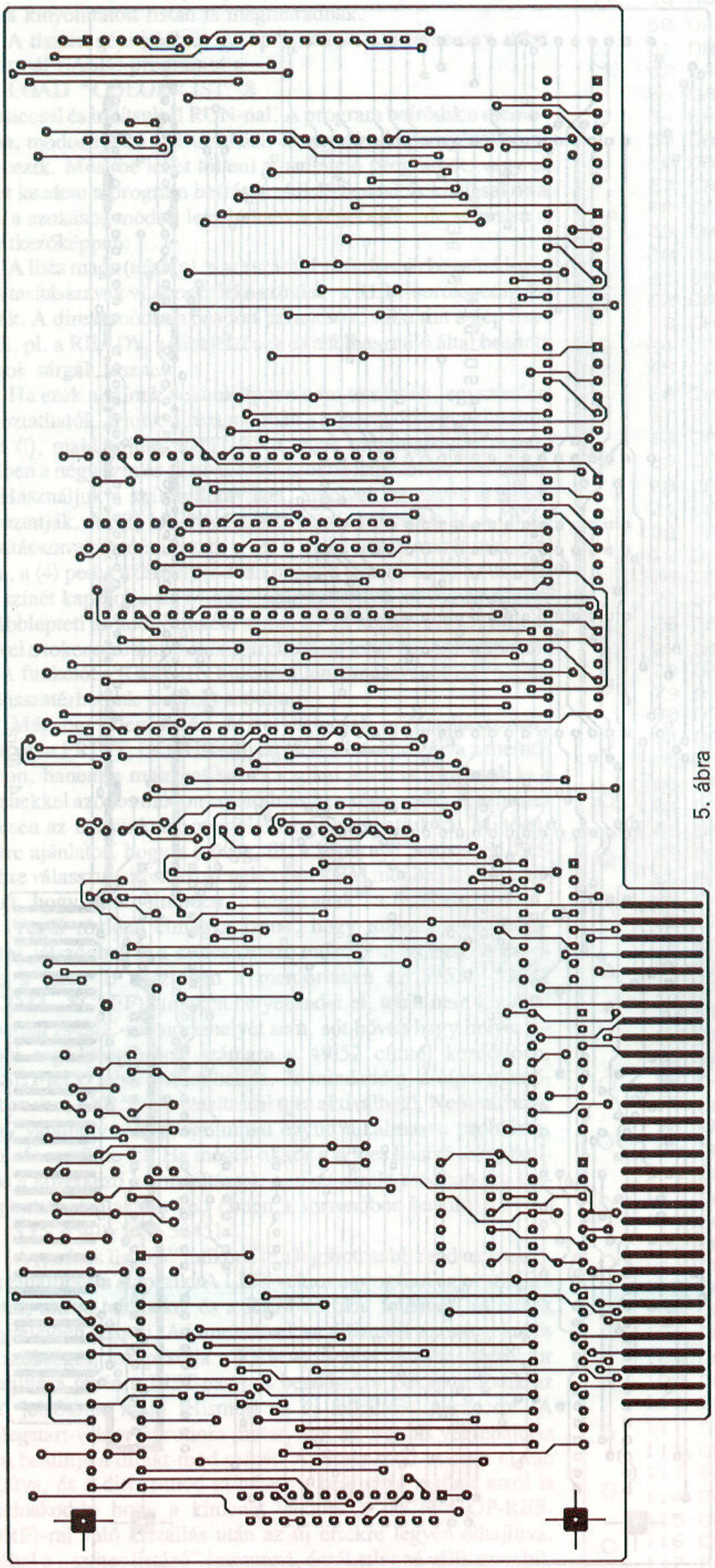
Hobbi ELEKTRONIKA

C-64 bővítések

ICTESZTER IBM XT/AT SZÁMÍTÓGÉPEKHEZ 2. rész



4. ábra



5. ábra

A program szolgáltatásai

A tesztelendő IC típusának meghatározása. Ez a típusszám, vagy típusszámok meghatározását jelenti és hogy az adott áramkör TTL vagy CMOS terheléssel működik-e jobban. A tesztelési idő függ a számítógép sebességétől kb. 1—2 másodpercig tart. Az eredmény a képernyőről egyértelműen leolvasható.

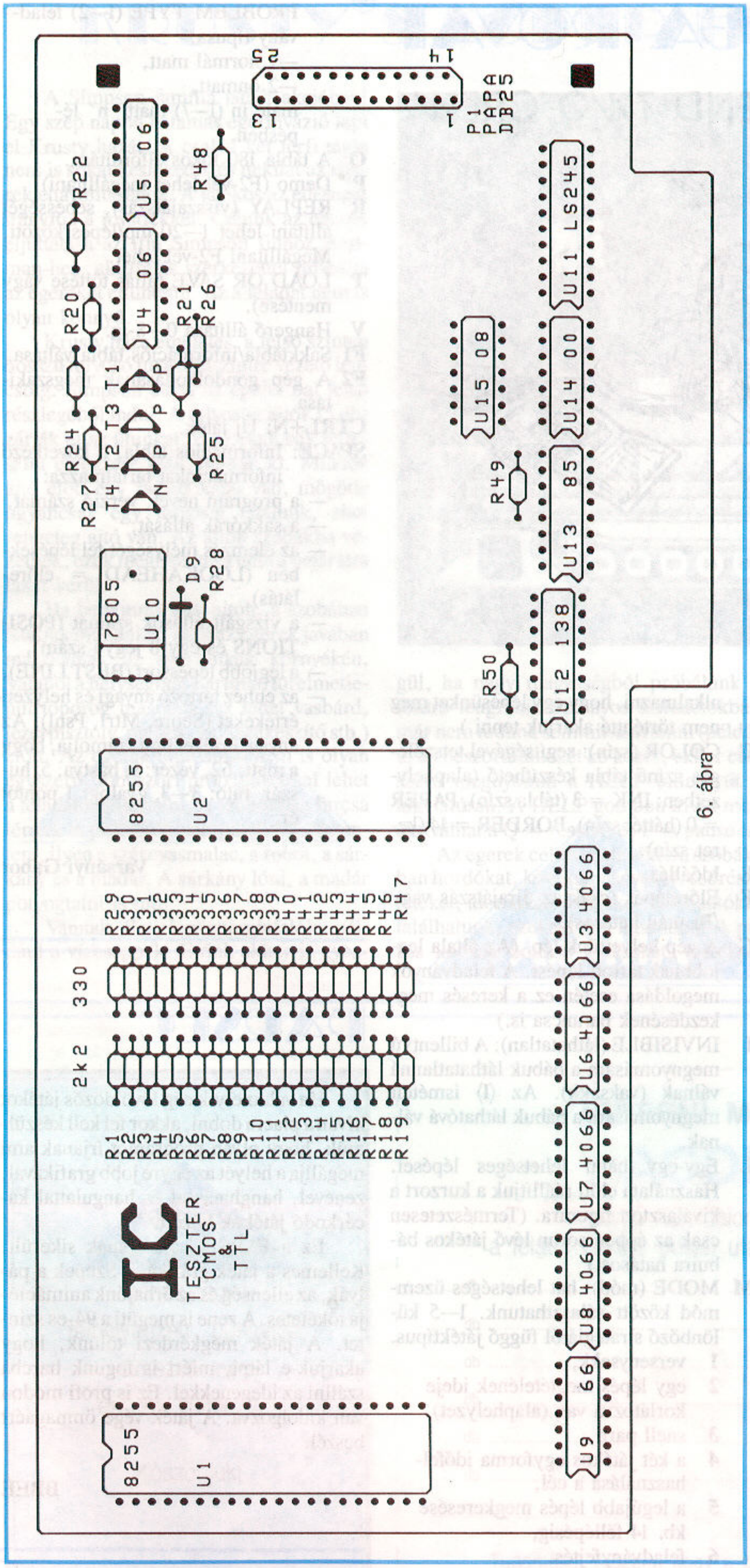
IC próba. A korábban megállapított típusú IC-t folyamatosan teszteli CMOS és TTL terheléssel és a tesztelés eredményét számlálókon jeleníti meg az IC nevével és rövid ismertetőjével együtt. A még behatóbb ellenőrzéshez a képernyő alján megjelenik a lábkiosztás, ahol a logikai hibát ,,B" betű jelöli. Így egy katalógus segítségével azt is meg lehet tudni, hogy az a kivezetés mire szolgált volna. Ez azért hasznos, mert ha fejlesztéskor több IC is ugyanazon a lábon ment tönkre, könnyebben megtalálja az elkövetett hibát. Katalógusnak nagyon jól használhatók azok a floppilemezek, melyek a szerző üzletében kaphatók. Ezekről képernyőn jelenítheti meg több ezer alkatrész adatát, olyan formában, mint ha nyomtatott katalógusból olvasná, s mindez négy lemezen elfér. Próbát nemcsak típusmegállapítás után lehet kérni, hanem a ,,lista az IC-ről" opcióban megjelölt típusról is.

Tesztelendő IC-k listája. Ez a tesztelhető IC-k nevét és rövid ismertetőjét tartalmazza, valamint ki lehet választani egy IC-t a próbához.

Ellenőrzés. A tesztelő áramkör működését ellenőrzi a leírtak szerint. Ha e menüpont kiválasztása előtt elfelejtette volna kivenni a tesztelendő IC-t, de a program felismeri annak típusát, akkor figyelmeztet, hogy az IC-t a teszterfoglalatból ki kell venni. Ha nem talál hibát, visszatér a főmenübe, ellenkező esetben az előzőekben leírt módon hibaüzenetet generál.

A teszter kezelése

A tesztelendő IC-t úgy kell behelyezni a teszterfoglalatba, hogy az 1. láb és a foglalat 1. csatlakozási pontja egybeessen. A doboz felirata ezt szemléletesen



6. ábra

mutatja. Utána le kell hajlítani a foglalat karját, majd lehet választani a menüből. A tesztelendő IC behelyezésekor nem világíthat a piros led, mert ilyenkor a tesztfoglalaton kinn van a tápfeszültség és ez tönkreteheti az IC-t. A CMOS IC-k között van olyan, amelyik a fordított behelyezéstől, vagy a nem megfelelő lábra adott tápfeszültségtől tönkremegy. A TTL IC-k minden forgatást és rossz helyre dugaszolást károsodás nélkül elviselnek.

A program futtatásához szükséges gépi feltételek:

IBM PC, XT, AT kompatibilis számítógép 64 K szabad RAM területtel.

CGA, EGA vagy VGA kompatibilis monitor.

Egy szabad kártyacsatlakozó hely.

A program betöltéséhez szükséges háttértároló.

Billentyűzet.

Az IC teszter árajánlata

Az IC teszter működő mintadarabja megtekinthető, kipróbálható, egységcsomagban vagy készre szerelve megvásárolható a szerző üzletében. Ára egységcsomagban 5900,— Ft. Készre szerelve, élesztve 9800, Ft. Külön vásárolható panel: ára 1800,— Ft.

Folyamatosan kapható egységcsomag, panel és részegység a C-újságban korábban megjelent cikkeimhez is. Ezek nagy része működés közben megtekinthető, kipróbálható. A vidéki olvasóknak segít a szerző levelező-egységcsomagküldő szolgáltatása. A megrendelt csomagot postán utánvéttel elküldöm.

Telefonon és levélben is rendelhet. A HOBBI ELEKTRONIKÁHOZ nem kell hosszú levél. Rendelését röviden, egyértelműen közölje. Az üzletben beszerezhetők a C-újság korábbi számai is.

Levélcím: HOBBI ELEKTRONIKA
1656 BUDAPEST Pf. 50.
Üzletcím: BUDAPEST VII.
Dózsa György út 16.
Jobbágy u. sarok.
Nyitva: H—P. 10—17-ig,
zárás után üzenetrögzítő.
Tel./Fax: 122-8892

C64 & AMIGA ROVAT

BUMBLE BEE & FRIEND TWO CREW



COLOSSUS 2.0 CHESS

A soron lévő játékost a bal alsó saroknégyzetben egy vízszintes vonal jelzi, (A1 vagy H8 mező), ez a vonal szolgál a lépések kijelölésére is.

A lépések megtétele úgy történik, hogy a kurzornyilak segítségével ráállunk arra a bábura, amellyel lépni akarunk, majd megnyomjuk a ⟨RETURN⟩ gombot — ennek a mezőnek a koordinátáját kiírja a gép — majd szintén a kurzornyilakkal megjelöljük a célmezőt és újra a ⟨RETURN⟩-t, — ezt a mezőt is kiírja, majd megteszi a kijelölt lépést, vagy 'ILLEGAL' hibaüzenetet ad. Ha eltévesztettünk valamit, a ⟨DEL⟩ billentyűvel törölhetjük.

A lépés kijelölése a 'SHIFT+A—H', majd a sor beírása 1—8 útján is történhet.

AZ EGYÉB BILLENTYŰK HATÁSA:

A ALTER POSITION (állás módosítása)
Ezen belül használatos jelek:
- **W** üres tábla,
- **S** színcsere,
- **K** király,
- **R** bástya,
- **B** futó,
- **C** mezőtörlés,
- **G** vissza a kiinduló helyzetbe,
- **M** lépésszámláló beállítása,
- **Q** vezér,
- **N** huszár,
- **P** gyalog,
- **E** visszatérés a játékba.

B Egy fél lépést visszalép. (Kétszer kell alkalmazni, hogy egy lépésünket meg nem történtté akarjuk tenni.)

C COLOR (szín): segítségével tetszőleges színű tábla készíthető (alaphelyzetben: INK = 3 (tábla szín), PAPER = 0 (háttér szín), BORDER = 14 (keret szín).

E Időállás.

F Előrelépés (csak az újrajátszás vagy ⟨B⟩ után hatásos).

G A gép helyettünk lép. (Az általa legjobbnak tartott lépést. A feladványok megoldása esetén ez a keresés megkezdésének parancsa is.)

I INVISIBLE (láthatatlan): A billentyű megnyomására a bábuk láthatatlanná válnak (vaksakk). Az ⟨I⟩ ismételt megnyomására a bábuk láthatóvá válnak.

L Egy-egy bábu lehetséges lépései. Használata előtt ráállítjuk a kurzort a kiválasztott figurára. (Természetesen csak az éppen soron lévő játékos bábuira hatásos.)

M MODE (mód): hat lehetséges üzemmód között választhatunk. 1—5 különböző stratégiától függő játéktípus.
- **1** versenysakk,
- **2** egy lépés megtételének ideje korlátozva van (alaphelyzet),
- **3** snell parti,
- **4** a két játékos egyforma időfelhasználása a cél,
- **5** a legújabb lépés megkeresése kb. 14 féllépésig,
- **6** feladványfejtés.

PROBLEM TYPE (1—2) feladvány típusa:
—1 normál matt,
—2 önmatt,
mate in (1—7) matt 'n' lépésben.

O A tábla 180 fokos elfordítása.

P Demo (F2-vel lehet megállítani).

R REPLAY (visszajátszás), sebességét állítani lehet 1—20 mp/lépés között. Megállítani F2-vel lehet.

T LOAD OR SAVE (állás töltése vagy mentése).

V Hangerő állítása 0—8-ig.

F1 Sakktábla/információs tábla váltása.

F2 A gép gondolkodásának megszakítása.

CTRL+N: Új játék.

SPACE: Információs tábla, a következő információkat tartalmazza:
- — a program nevét, verzió számát,
- — a sakkórák állását,
- — az elemzés mélységét fél lépésekben (LOOKAHEAD = előrelátás),
- — a vizsgált állások számát (POSITIONS és egy 8 jegyű szám),
- — a legjobb lépéssort (BEST LINE),
- — az ehhez tartozó anyagi és helyzeti értékeket (Score, Mtrl, Psnl). Az anyagi értéket úgy számolja, hogy a matt: 62, vezér: 9, bástya: 5, huszár, futó: 3—3, gyalog: 1 pontot ér.

Varsányi Gábor

PARSEC

Ha valamelyik cég lövöldözős játékot kíván a piacra dobni, akkor fel kell készülniük, hogy olyan programot írjanak ami megállja a helyét az egyre jobb grafikával, zenével, hanghatással és hangulattal kacérkodó játékok között.

Ez a PARSEC alkotóinak sikerült. Kellemes a játék grafikája. Szépek a pályák, az ellenség és az űrhajónk animációja tökéletes. A zene is megüti a 94-es szintet. A játék megkérdezi tőlünk, hogy akarjuk-e látni, miért is fogunk harcba szállni az idegenekkel. Ez is profi módon van kidolgozva. A játék vége önmagáért beszél.

BBEE

KRUSTY'S SUPER FUN HOUSE

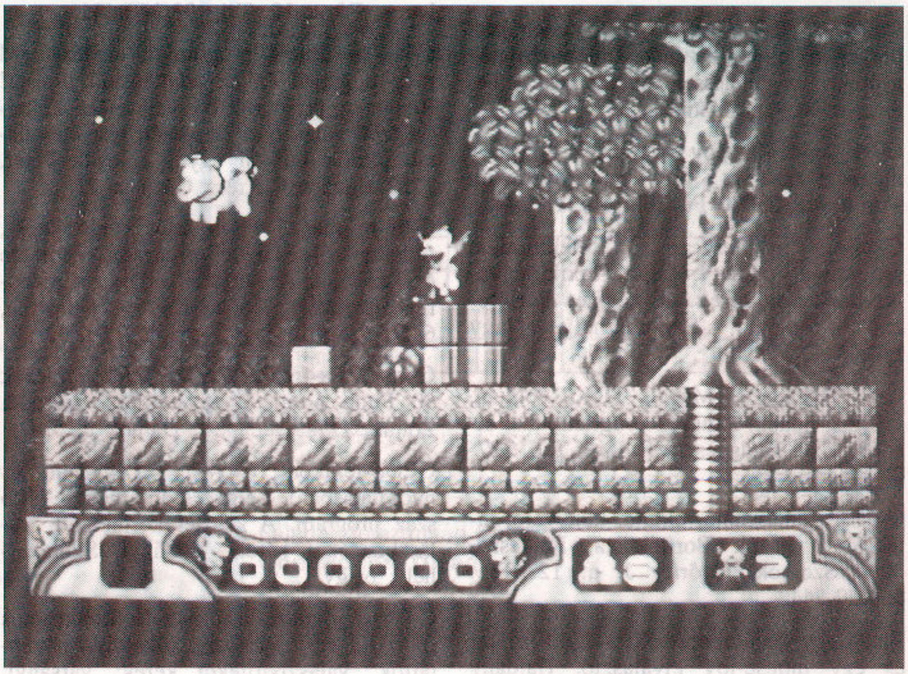
A Simpson família ismét akcióban! Egy szép napon hatamas egérinvázió lepi el Krusty házát. A család két férfi tagja nem is sokat késlekedik és nekilát az egerek eltávolításához. A játékos az édesapát irányítja a joystickkel. Célunk az egerek eljuttatása az ifjú Simpson fiúhoz Bartman-hez, aki tréfás szerkezettel próbálja az egereket eltüntetni. Ez a feladat nem is olyan könnyű.

Krusty háza emeletes, a felső szinten hosszú folyosó fut a földszintre vezető lépcsőig. Simpson bácsi az épület bal felső részlegéből indul. A folyosón ajtók (5 db) zárják el az útunkat, ezek csak egy teljes szint elvégzése után nyílnak ki. Minden ajtó előtt egy rácsos kapu van, mögötte ugyancsak egy folyosót találunk, ahol rengeteg ajtó van. Az ajtók szobákba vezetnek, ezek megtisztítása után a bejáratra lakat kerül.

Ha benyitunk egy ajtón, a szobában találjuk magunkat, ahol az egerek javában mászkálnak az egérlyukak környékén. Valahol a helységben Bartman türelmetlenül toporog masinája (ez lehet vasbárd, lézerpisztoly, pumpa, galuskakészítő stb.) előtt. Az egerek eljuttatása nem is olyan egyszerű, többszöri próbálkozással lehet a feladatot teljesíteni. A szobákban furcsa lények is próbálják energiánkat csökkenteni, ilyen a szárnyasmalac, a robot, a sárkány és a madár. A sárkány lőni, a madár potyogtatni is tud.

Vannak élettelen energiafalók, például a vízcseppek. Erőnk akkor is gyengül, ha nagy magasságból próbálunk a földre leugrani. Bizonyos szituációkban már nem tudunk semmit sem tenni (például ha beszorulunk két kő közé), ekkor célszerű megnyomni a HELP billentyűt a klaviatúrán. A HELP gombon kívül még aktiválhat a Q = restart, és P = pause is.

Az egerek célravezetéséhez a szobákban hordókat, köveket, csöveket, csőrészleteket, idomokat, üvegeket, dobbantókat találhatunk, ezeket lehet használni. A piros kockák bonusz kincseket, életeket, golyókat tartalmazhatnak. A képernyő alsó részében baloldalt láthatjuk a felszedett tárgy képét, jobboldalt a töltények és az életek számát. Lőni a tűzgombbal tudunk. Az elvégzett pályák végén kódszámokat kapunk, ezeket a címképnél lehet beírni, ilyen a NELSON, PATTIE, MRPLOW, MAGGIE. Beírásuk a joy fel és le mozgatásával lehetséges. Akik a Lemmings I, II valamelyikével találkoztak biztosan megkedvelik ezt a programot is.



AMIGA

C-64-re 4800 játék, 1600 felhasználói programból válogathasz. Válaszborítékért tájékozató. Halász, 6001 Kecskemét, Pf.: 512.

C-64-es, PLUS/4-es, C-16-os legújabb programok olcsón kazettán, lemezen. írjatok géptípus megjelöléssel. Akció! Bende Ferenc, 8500 Pápa, Attila u. 5/b.

C-64 eladó + floppy + lemezek + tartó + magnó + kazetták + cartridge + joystickok + könyvek 25000 Ft. Tel.: 92/318-343.

C-64-re! nyáktervező és elektronikai programok, poke- és GEOS-lexikon. Válaszborítékot kérek. A készlet korlátozott! Varsányi, Szombathely, Nagy László u. 11.

C-64 tápegység, Maxel, TDK, Pelikán, Proflex, nonam-lemezek programokkal eladók. 1224 Budapest, VII. u. 4. Tel.. 227-22-60.

C-64-re felhasználó, oktató, könyvelő, játékok 5 Ft-tól. Borítéklista. Földesné, 5000 Szolnok, Győrffy J. u. 12. Tel.: 56/420-544.

Eladó egy Commodore MPS-1230 printer és egy mouse-joy kiválasztó. Hajdány Ádám, 1145 Bp., Újvilág u. 39/b.

C-64 gyorstöltő, fejbeállító 700, FINAL III, ACTION MK VI. 2900, SIMON'S + HELP 1500 Ft. Mikroklub, 8100 Várpalota, Pf. 65. Tel.: 88/371-439.

Akarsz programot C+4-re és C-64-re? Ha igen, küldj 1 db lemezt. Bognár Péter, 8900 Zalaegerszeg, Madách út 6. III/10.

Itt a THE BEST SOFT! Superolcsó itt minden. Írj! THE BEST SOFT, 7400 Kaposvár, Hermann u. 60.

576-ban megjelent és egyéb C-64-es programok eladása lemezen, kazettán. Lemezes programok kazettán. Írj te is minél hamarább! Deutsch Szabolcs, 8749 Zalakaros, Fő u. 19. Tel.: 93/340-481.

C64, és Plus/4 programok eladása. Írd meg a géped és az adattárolód típusát! Rengeteg program, bő választék. Felbélyegzett válaszboríték esetén részletes listát küldök infóval együtt. Novák Péter, 1132 Budapest, Kresz Géza u. 21.

C-64-re már 3 Ft/db ártól eladó játék-, felhasználói- és egyéb programok lemezre és kazettára. A választék óriási. Válaszborítékért listát küldök. Kovács Krisztián, 6750 Szeged-Algyő, Bartók B. u. 6.

C-64 és C+4 kazetták 200 Ft/db, lemezek 50 Ft/db áron! Listát küldök. Csernáth Zoltán, 1171 Budapest, Dalnok u. 5.

GEOS-lexikon C-64-re! Lemezen és nyomtatásban! Leírás gyűjtemény. Több száz program. A készlet korlátozott. Válaszborítékot kérek: Varsányi, Szombathely, Nagy L. u. 11.

C-64-re jogtiszta szoftverek. Elektromos (NYÁK, stb) tervezők, alkatrésznyilvántartók. Válaszborítékot kérek: Varsányi, Szombathely, Nagy L. u. 11.

Eladó C-64/II + 1541 floppy + magnó + programok + 2 joy = 23 000 Ft. Dávid Zsolt, Győrújbarát, Tel.: 96/355-283.

C-64-hez profilváltás miatt eladó 1 db eprombank, 1 db epromégető és 2 db játékcartridge. Pázsit, 5650 Mezőberény, Várkonyi u. 1.

Keresem megvételre vagy cserébe GEOS mindenkinek C-64-re című könyvet. Váradi Ernő, 2151 Fót, Pf.: 43.

C-64 floppy — magnó — joy + Siskosha nyomtató, szakkönyvek, programok eladók. Szolnok, Tel.: 56/373-580 délután.

C+4 magnóval–szakkönyv–joy–programok eladó. Szolnok, Tel.: 56/373-580 délután.

C-64-re oktató, számlázó és egyéb programok, leírások, játékok 5 Ft-tól. Borítékért lista. Földesné, Szolnok, Győrffy J. u. 12. I/7. Tel.: 56/420-544.

Action Replay MK5 2900 Ft, MK7 + Atomic egyben 4500 Ft. Többféle Amigás kiegészítő kapható. Derko, 1399 Bp. Pf.: 701/679. Tel.: 291-23-18.

Helló! Akarsz C-64-re olcsón programokat? Akkor írj! THE BEST SOFT, 7400 Kaposvár, Hermann O. u. 60.

SZJA '93 programok C16, +4, 64-re lemezen, kazettán eladó, utánvéttel 480 Ft, Bagdács Lajos, 4130 Derecske, Dózsa Gy. u. 15.

Szeretnél 1250 db lemezt C-16 +4-re? 13 Ft-os bélyegért tájékoztató. Soós Viktor, 7355 Nagymányok, Katona u. 11.

C-16, +4-re programokat adok–veszek–cserélek, 5 Ft/db, válaszboríték! Soós Viktor, 7355 Nagymányok, Katona u. 11.

C-16, +4-re 280 játék örökélet poke-jai, indítási címei. Ára: 100 Ft. Soós Viktor, 7355 Nagymányok, Katona u. 11.

Mejelent C-64-re a legújabb szöveges, magyar kalandjáték: A tiltott város. 35 Kbyte szöveg 5 Kbyte kép. Lemez + kézikönyv. A programokért cserébe küldj 3 db 2S-2D-s lemezt + egy 30 Ft-os bélyeget. Kovács Tamás, 8500 Pápa, Vajda P. ltp. 22.

Kedves olvasónk! A1000, A500, A500+, A600, A1200, és egyéb Amiga számítógépekre, valamint C64, C128-re játék- és felhasználói prg-k, valamint PD demók megrendelhetők. Vízhatlan csomagolás, kedvezmények, számla. Bélyeggel ellátott válaszborítékért részletes tájékoztatót és programlistát küldünk. Amigán a lista mágneslemezen is kérhető egy 3.5-ös lemez, és 30 Ft-os bélyeg ellenében. A borítékokat géptípusonként a következő címekre lehet feladni:
AMIGA: FRIEND TWO CREW, 1399 Bp., Pf.: 701/55. Tel.: 135-12-89. (Ügyelet naponta 16—20 óra között)
C-64: BUMBLE BEE, 1116 Bp., Sárkány u. 44. Tel.: 162-77-24.

Kedves Olvasó!
Lehetőséget biztosítunk mindenkinek cikkei, leírásai, ismertetői újságban történő közlésére. Várunk játék, felhasználói és demo programokról készült írásokat, játékkódokat, tippeket, fortélyos fogásokat, amelyek Amiga és Commodore 64/128-as számítógépekhez készültek. Minden közlésre alkalmas írás szerzője egy csomag márkás audiókazettát kap ajándékba. A rovat levélcíme: FRIEND TWO CREW, CIKK Budapest Pf.: 701/55.

PÖTYÖGŐ SZOLGÁLAT

307	C64	91/ 9	Adventure - Sz	130.-
308	C64	91/ 9	Morze Oktató B.	130.-
309	+4	91/ 9	Memory Game Plus	100.-
310	C64	91/ 9	*Capitaly	130.-
311	C64	91/ 9	*Gengszterek	60.-
312	C64	91/ 9	*Küldetés 2000	100.-
313	C64	91/ 9	Itt a $ hol a $	50.-
314	C64	91/ 9	Hewing	50.-
315	+4	91/10	Mikrobi	50.-
316	+4	91/10	Nest of Fleas	190.-
317	+4	91/10	Adventure - F	100.-
319	C64	91/10	Gondolkozó	100.-
320	C64	91/10	Memori - B	130.-
321	+4	91/10	Matekastély	100.-
322	C64	91/11	Lottó	80.-
323	C64	91/11	Totó	80.-
324	C64	91/11	Német gyakorló	60.-
325	+4	91/11	Beszélö szótár	130.-
326	+4	91/11	File examiner	60.-
327	C64	91/11	Sprite szerkesztö	50.-
328	C64	91/11	Egyszinü Sprite	50.-
329	C64	91/12	Rulett javitás	50.-
330	C64	91/12	Tözsde	130.-
332	C64	92/ 1	Kalandlap	80.-
333	C64	92/ 1	Levéliró	80.-
334	C64	92/ 1	Képernyömaszkoló	130.-
335	C64	92/ 1	Sprite Kezelö	100.-
336	C64	92/ 1	SI - OPART	40.-
337	C64	92/ 2	Patience Quartet	130.-
338	C64	92/ 2	Monopoly	130.-
339	C64	92/ 2	Akasztófa	80.-
340	C64	92/ 2	Karaktertervezö	60.-
341	C64	92/ 3	String Array Manager	100.-
342	C64	92/ 3	Sprite Basic	50.-
343	C64	92/ 3	Screen Help	50.-
344	C64	92/ 3	Gyors validate	60.-
345	C64	92/ 3	BASIC SYNTAXER I.	190.-
346	C64	92/ 4	Tologató C64-re	100.-
347	C64	92/ 4	ABC rendezö	100.-
348	C64	92/ 5	*Matek + Római számok	100.-
349	+4	92/ 6	Tesztkészitö + értékelö	100.-
350	C64	92/ 3	BASIC SYNTAXER II.	190.-
351	+4	92/ 7	Három játék	110.-
352	+4	92/ 7	Szökés	80.-
353	C64	92/ 7	Toto	150.-
354	C64	92/ 7	DIGIT analizis program	110.-
355	+4	92/ 7	SCREEN DUMP	150.-
356	C64	92/ 9	UNISEQ filekezelö	130.-
357	C64	92/ 9	Graphic Copy	60.-
358	C64	92/10	Printfox Resizer	130.-
359	C+4	92/11	Tologató II.	150.-
360	C+4	92/12	Sprite Plus-4-en	130.-
361	C64	92/12	Változólista	150.-
362	C64,+4	93/ 2	Szomszéd játék	200.-
363	C64	93/ 2	Táblás játék	200.-
364	C64	93/ 2	SEDA	150.-
365	C64	93/ 2	415 * 200	120.-
366	+4	93/ 2	OTP hitel	100.-
367	C64	93/ 3	Catalog-Check(CTC)	150.-
368	C64	93/ 3	Pattogó golyók	200.-
369	C64	93/ 3	Beszoritós (DO-GUTI)	150.-
370	C64	93/ 3	*Zsirozás	200.-
371	C64	93/ 4	Attrib 64	50.-
372	C64	93/ 4	Táblázatnyomtató	80.-
373	C+4	93/ 4	Boritékkészitö	80.-
374	C+4	93/ 4	TV-óra	50.-
375	C64	93/ 4	Mini Domino	150.-
376	C64	93/ 4	Megathello	120.-
377	C64	93/ 4	Reversi	120.-
378	C64	93/ 4	Tengeri csata	120.-
379	C64	93/ 5	Videokazetták	80.-
380	C+4	93/ 5	DIN-karakterek	80.-
381	C64	93/ 5	DIRYDIRY	80.-
382	C64	93/ 5	Birodalom	150.-
383	C64	93/ 6	Csöamöba	100.-
384	C+4	93/ 6	Függvényábrázolás	150.-
385	C64	93/ 6	Amöba - E	100.-
386	C64	93/ 6	Body Building	60.-
387	C64	93/ 7	Colour BASIC	80.-
388	C64	93/ 7	Directory kezelö	60.-
389	C64	93/ 7	ötmezös KONO	80.-
390	C64	93/ 7	Fogócska	80.-
391	C+4	93/ 7	Klick - Klack	120.-
392	C64	93/ 9	Orarend	60.-
393	C64	93/ 9	Malomjáték	80.-
394	C+4	93/10	Magyar nyomtatás	120.-
395	C64	93/10	Pettyes szfinx	60.-
396	C64	93/11	Ku Damm óra	60.-
397	C64	93/11	Elásott kincs	80.-
398	C64	93/11	Turbo Sort	80.-
399	C64	93/12	Telefonkönyv	80.-
400	C64	93/12	Színcserélö	80.-
401	C64	94/ 1	Sprite készitö	80.-
402	C64	94/ 1	Amöba	80.-
403	C64	94/ 1	3 hasábos nyomtatás(TCT)	100.-

A *-gal jelölt programok csak lemezen rendelhetök !

Kérem, hogy a **Commodore Újság** legközelebbi számában jelenjen meg a következő szövegű apróhirdetés:

(minden kockába egy betűt írjon)

Tagsági szám: ....

Feladandó az újság címére:
Commodore Újság
Budapest, 1388. Pf. 86.

Alulírott megrendelem a következő programokat a PÖTYÖGŐSZOLGÁLAT-tól:

PROGRAM SORSZÁMA	ÁRA
1.	,– Ft
2.	,– Ft
3.	,– Ft
4.	,– Ft
5.	,– Ft
6.	,– Ft
7.	,– Ft
8.	,– Ft
9.	,– Ft
Összesen: db	,– Ft

A programokat a SZOLGÁLAT által biztosított
lemezre (99 Ft/db)
kazettára (40 Ft/db)
az általam küldött adathordozóra kérem.
(Kérjük a megfelelő szöveg aláhúzását!)
Postaköltség + munkadíj
A fizetés módja: személyesen — csekken — utánvéttel
(Kérjük a megfelelő szöveg aláhúzását!)
ÖSSZESEN:

A megrendelő neve: ....

Címe: ....

A NOVOTRADE SZERVÍZ Kft. az alább felsorolt szervízeiben mindenféle szervízszolgáltatás munkadíjából 10% kedvezményt ad az egyesületi tagoknak.

1083 Budapest, Szigony u. 8.	Tel.: 1343-153
3525 Miskolc, Fazekas u. 1–3.	Tel.: 46/321-488
5600 Békéscsaba, Bartók B. u. 37.	Tel.: 66/327-195
6724 Szeged, Csongrádi sugárút 76.	Tel.: 62/493-185
9700 Szombathely, Szalonak u. 31.	Tel.: 94/314-519

NOVOTRADE SZERVÍZ Kft.

Igazolás: a javítandó berendezés leadásakor egyesületi igazolvánnyal. A kedvezmény többször is igénybe vehető.

Az Országos Commodore Egyesület szolgáltatásai

Egyesületi tagoknak 20% kedvezmény:

VC—20 memóriabővítés 3—27 kByte-os:	kiépítéstől függő
C—16, C—116 memóriájának bővítése 64 kByte-ra:	3500 Ft
C—16 belső 16 kByte-os EPROM bővítés:	1450 Ft
C—16 belső 32 kByte-os EPROM bővítés:	2900 Ft
C—16 belső 8 kByte-os SOFT—ROM bővítés:	2800 Ft
C—16 belső 32 kByte-os SOFT—ROM bővítés:	4000 Ft
C—16 8 kByte-ról 32 kByte-ra átalakítás:	2000 Ft
C—16 és 1541 kompatibilis lemezegység párhuzamosítása:	3200 Ft
SOFTROM modul 32K, kikapcsoláskor sem felejt C-16, C-116, +4	5000 Ft
FÉK C—16, C—116, +4 potméteres sebességváltoztatás 0%-tól 100%-ig fokozatmentesen	2000 Ft
TTL IC-teszter (Cartridge+lemezen a program)	4300 Ft
+4, C—16, C—116 UNI—ROM modul különféle kiépítésekben:	
— 8 kByte SOFT—ROM	3400 Ft
— 16 kByte SOFT—ROM	4000 Ft
— 8 kByte SOFT—ROM 16 kByte EPROM	4400 Ft
— 16 kByte SOFT—ROM 16 kByte EPROM	5000 Ft
— 16 kByte EPROM	2200 Ft

Egyesület tagoknak 30% kedvezmény:

Speeddos (átkapcsolható) operációs rendszer beépítése (C64 átalakítás, lemezegység átalakítás + párhuzamos kábel)	5000 Ft
1541 kompatibilis lemezegységbe elektronikus lemezlyukasztó beépítése	900 Ft
PAGEFOX magyar ékezetes kiadvány-szerkesztő cartridge (a teljes A/4-es oldal kinyomtatásához 640 pont/soros nyomtató szükséges minimum, pl. Citizen 120D)	5500 Ft
FASTLOAD cartridge (lemezes gyorstöltő, másoló, monitor)	1500 Ft
TTL IC-teszter cartridge + program	4300 FT
288/256 Kbyte-os eprombank (vezérlő eprommal)	5000 Ft
Eprométegető (2716-tól 27512-ig)	5000 Ft
C64-hez tároló oszcilloszkóp	8000 Ft
C64-bővítő-port elosztó (egyszerre 4 db cartridge lehet a gépben, melyeket gombnyomásra lehet kapcsolni)	7500 Ft
C64 USER — CENTRONICS nyomtatókábel (GEOS kábel)	1500 Ft
256 K RAM-diszk (256 Kbyte RAM-mal)	14000 Ft
256 K RAM-diszk (64 Kbyte RAM-mal)	9000 Ft
2×64 Kbyte-os cartridge igény szerinti programokkal feltöltve	4.300 Ft
64 Kbyte-os cartridge igény szerinti programokkal feltöltve	3000 Ft
Képújság (teletext) dekóder C-64-re	10000 Ft
1764-es RAM bővítő GEOS-hoz is használható	11250 Ft
Epromok programozása meglévő programokkal, vagy saját hozott programok beégetésével 2716-tól 27512-ig az eprommal együtt egységesen	700 Ft

Árainkat az alkatrészárak változásai befolyásolhatják.

A fenti bővítések megrendelhetők levélben az O.C.E. címén, valamint személyesen a havonta rendezendő klubdélelőttön, ahol rendszeres bemutatót is tartunk.

Postázás esetén 100 Ft postaköltség kerül felszámításra.

A kedvezmény igénybevételéhez az O.C.E. tagsági igazolvány bemutatása szükséges.

Ára: 88 Ft

ACOMP

SZÁMÍTÁSTECHNIKAI KFT.

1125 Budapest, Királyhágó utca 2. — **Tel.: 156-6790** — **Fax: 251-2385**
1135 Budapest, Szent László út. 74./A. — **Tel.: 149-6165** — **Fax: 251-2385**
1191 Budapest, Katica utca 9. — **Tel.: 280-4267** — **Fax: 177-9419**

Nyitva: hétfőtől péntekig: 9—17 óráig, szombaton 9—13 óráig.

Commodore Amiga 500 Plus	29 990 Ft
Commodore Amiga 600	29 990 Ft
Commodore Amiga 1200	52 990 Ft
Commodore Amiga 1200 Desktop Dynamite	59 990 Ft
Commodore Amiga CD-32	49 990 Ft
A 600 / A 1200-hoz 40 MB HD kábellel	21 990 Ft
60 MB HD kábellel	24 990 Ft
80 MB HD kábellel	28 990 Ft
120 MB HD kábellel	36 990 Ft
Commodore Amiga 4000/040/6MB/00MB	309 000 Ft
+ 4 MB RAM modul	29 990 Ft
Commodore Amiga 4000/030/4MB/00MB	199 000 Ft
Commodore 1942 Multisync Stereo Monitor	59 990 Ft
Commodore 1084s Stereo-Color monitor	33 990 Ft
Commodore 1085s Stereo-Color monitor	33 990 Ft
Commodore A-520 TV-Modulator (Amigával)	4 990 Ft
Commodore A-520 TV-Modulator	6 990 Ft
Commodore Amiga – EuroScart kábel	1 290 Ft
Commodore C-64 TERMINATOR SET	14 990 Ft
Commodore 1541/II Floppy drive (C-64-el)	13 990 Ft
Commodore Datassette	2 990 Ft
Commodore MPS 1230 printer	23 990 Ft
Commodore A570 CD-ROM driver + 3 CD	16 990 Ft
Sega Megadrive + 1 pad + 1 játék	15 990 Ft
512 Kb órás memóriabővítő	4 990 Ft
2.0 MB órás memóriabővítő	9 900 Ft
1.0 MB-os chip bővítő A500 Plus-ba	6 990 Ft
1.0 MB-os órás chip bővítő A600-ba	7 990 Ft
1.0 MB-os órás fast bővítő A1200-ba	16 990 Ft
2.0 MB-os órás fast bővítő A1200-ba	23 990 Ft
4.0 MB-os órás fast bővítő A1200-ba	37 990 Ft
8.0 MB-os órás fast bővítő A1200-ba	62 990 Ft
2.0 MB PCMCIA bővítő A600-ba, A1200-ba	19 990 Ft
Sound Blaster PRO-2 Deluxe	16 900 Ft
Sound Blaster 16 BASIC	21 900 Ft
Sound Blaster 16 ASP Deluxe	33 900 Ft
Noris Porvédő Amiga 600	990 Ft
Noris Porvédő C-64 II.	790 Ft
Noris Porvédő C-64 I.	150 Ft
NoName 3.5" DSDD lemez	550 Ft
NoName 3.5" DSHD lemez	790 Ft
NoName 5.25" DSDD lemez BULK	200 Ft
Maxell 3,5" MF2-HD lemez	1 390 Ft
Polaroid 3.5" DSDD lemez	690 Ft
Polaroid 3.5" DSHD lemez	1 190 Ft
Polaroid 5.25" DSDD lemez	450 Ft
Polaroid 5.25" DSHD lemez	650 Ft
Profex 3.5" DSDD lemez (11 db/Form)	690 Ft
Profex 3.5" DSHD lemez (11 db/Form)	990 Ft
Profex 5.25" DSHD lemez (11 db/Form)	390 Ft
Action Replay MK VI. Pro (C-64) + könyv	5 900 Ft
C-64 midi szoftverrel	6 500 Ft
Swifty Amiga Mouse (3 gombos)	2 500 Ft
Képdigitalizáló + RGB splitter	13 900 Ft
Mouse — Joystick automatikus kiválasztó	2 500 Ft

Real Time Clok Amiga 1200-hoz	2 500 Ft
Midi Amiga Interface	3 500 Ft
Handyscanner fekete/fehér Amigához	17 990 Ft
Stereo hangdigitalizáló Amigához	6 990 Ft
Sound Enhancer Amigához	3 500 Ft
Trackball Amigához	3 500 Ft
Rockey Advanced Video Keying for Amiga	24 900 Ft
G. V. P. G-Lock genlock S-VHS	59 900 Ft
G. V. P. A1230-II/50/00/4 + DPP-II	119 900 Ft
G. V. P. A1291 SCSI Option	14 900 Ft
G. V. P. Impaet Vision 24/GT–PAL	198 900 Ft
1.76 MB HD külső drive Amigához (OS2+)	16 990 Ft
3.5" Amiga slim külső drive	9 990 Ft
2.5"—2.5" Hard disk kábel	2 990 Ft
2.5"—3.5" Hard disk kábel	2 990 Ft
Mouse pad	290 Ft
Noris üveg 14" földelhető monitorfilter	1 290 Ft
Noris MB 80 3,5" lemeztartó	490 Ft
Noris DB 100 5,25" lemeztartó	490 Ft
Amiga Magazin (német) újság	490 Ft
Power Play (német) újság	490 Ft
Beyond the Minds Eye (Computer Animáció)	3 000 Ft
Joystickok	
Quickshot QS – 113 IBM	1 090 Ft
Quickshot QS – 123 Warrior IBM	1 290 Ft
Quickshot QS – 146 Intruder5 IBM	2 890 Ft
Quickshot QS – 151 Aviator5 IBM	3 590 Ft
Quickshot QS – 172 Raider5 IBM	1 650 Ft
Quickshot QS – 189 Python5 IBM	1 290 Ft
Quickshot QS – 191 Starfighter5	1 290 Ft
Quickshot QS – 101 I.	550 Ft
Quickshot QS – 111A II. Turbo Mikrokap.	990 Ft
Quickshot QS – 128 Maverick1	1 390 Ft
Quickshot QS – 130F Python1	1 090 Ft
Quickshot QS – 137F Python1M Mikrok.	1 150 Ft
Quickshot QS – 155 Aviator1 Mikrokap.	2 890 Ft
Quickshot QS – 149 Intruder1	2 690 Ft
Quickshot QS – 130N Pyton2 Nintendo	990 Ft
Dynamics Competition Pro IBM	3 990 Ft
Dynamics Competition Pro 5000	1 190 Ft
Dynamics Competition Pro 5000 Mini	1 690 Ft
Dynamics Competition Star Mini	2 190 Ft
Dynamics Competition Special Mini	1 690 Ft
Dynamics Competition Transparent Mini	1 690 Ft

Commodore CD-32 játékok. Egységáron: 4 990 Ft
Nigel Mansell's World Championship, The Labirinth of time, Zool, Morph, Alfred Chicken, Sensible Seccer, Deep Core, Liberation, D-Generation, Pinball Fantasies, Mean Arenas, Whale's Voyage, Troll, James Bond 2, Overkill & Lunar-C, Castles II., Pirates Gold, Dangerous Streets

SEGA MEGADRIVE JÁTÉKOK,
SEGA MEGA CD II., GAME GEAR
Árakról telefonon érdeklődjön!

Áraink mindenkori változtatásának a jogát fenntartjuk!

Áraink az 1 év garanciát és az ÁFA-t tartalmazzák!

SZERENCSEJÁTÉK RT